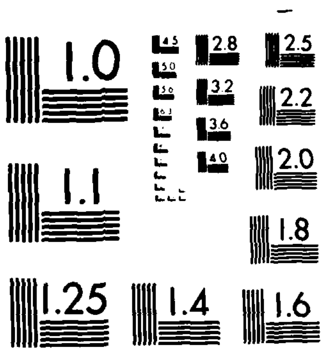
AD-A151 765 COMPUTER MODELING OF VIBRATIONAL ENERGY LEVELS OF POTENTIAL LASER CANDIDA..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P H OSTDIEK 1/2

UNCLASSIFIED DEC 84 AFIT/GEP/PH/84D-6 F/G 20/8 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

1

COMPUTER MODELING OF VIBRATIONAL ENERGY LEVELS OF POTENTIAL LASER CANDIDATES (DIATOMIC MOLECULES)

THESIS

Paul H. Ostdiek
Captain, USAF

AFIT/GEP/PH/84D-6

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GEP/PH/84D-6

COMPUTER MODELING
OF VIBRATIONAL ENERGY LEVELS
OF POTENTIAL LASER CANDIDATES
(DIATOMIC MOLECULES)

THESIS

Paul H. Ostdiek
Captain, USAF

AFIT/GEP/PH/84D-6

AFIT/GEP/PH/84D-6

COMPUTER MODELING OF VIBRATIONAL ENERGY LEVELS OF POTENTIAL LASER CANDIDATES (DIATOMIC MOLECULES)

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

Paul H. Ostdiek, B.S.
Captain, USAF

December 1984

Approved for public release; distribution unlimited

Acknowledgements

I am indebted to the Physics faculty for their efforts to prepare me for this thesis. Particularly, I wish to thank Dr. Ernest A. Dorko and Dr. Donn G. Shankland. Their guidance, experience, and ideas are greatly appreciated. The approach and numerical treatment presented in this work were developed by Dr. Shankland.

Finally, I wish to acknowledge the love and support given me by my wife Karen and son David these past eighteen months.

Paul H. Ostdiek

Contents

List of Figures

List of Tables

Abstract

This thesis developed a finite element solution of the Schrodinger wave equation. This technique is used by a computer program to calculate the energy levels and wave functions of a diatomic molecule for a particular potential energy model. The potential energy model is a function of a set of parameters which a non-linear minimization routine varies before solving the wave equation. This is done in an iterative manner until the calculated energy levels agree in a least squares sense with the observed energy levels. Then the transition probabilities (Franck-Condon factors) between the wave functions are calculated by another program developed for this thesis. Finally, two programs were written to determine the energy levels observed in spectroscopic data. One uses Dunham coefficients and the Dunham equation while the second uses a least square fit to the data directly.

The four programs were tested and appear to work correctly. The numeric solutions were compared with the analytic solutions of the single harmonic oscillator. The lowest 25 energy levels agreed to within 0.005% accuracy while their wave functions appear to agree to within 0.40% accuracy.

COMPUTER MODELING OF VIBRATIONAL ENERGY LEVELS
OF POTENTIAL LASER CANDIDATES (DIATOMIC MOLECULES)

1. Introduction

Background

The evaluation of the lasing potential of diatomic molecules is simplified by accurate knowledge of the molecule's energy as a function of internuclear distance. This knowledge is used to generate wave functions from the Schrodinger wave equation which describe the molecule in a particular quantum state. These wave functions are [illegible] in turn to calculate the probability that the molecule will change from one state to the other. This probability, or Franck-Condon factor, eases the correlation of spectroscopic data.

AFIT began an effort to develop a set of computer routines capable of calculating these Franck-Condon factors for diatomic molecules in 1982. The central program of this set was acquired from Dr. C.R. Vidal (Max Planck Institut fur Extraterrestriche Physik) (26). This program uses the semi-classical Rydberg-Klein-Rees (RKR) procedure and an Inverse Perturbational Analysis (IPA) to generate the molecule's potential energy curve. Capt. L.L. Rutger modified Vidal's program to run on the CDC CYBER computer system in his thesis effort (March 83). Rutger also wrote a program to generate

the set of molecular constants used as input to the RKR-IPA code (20). Then, in another thesis project (December 83), Capt. J.J. Pow completed the program set by creating programs to plot the energy curves and calculate the Franck-Condon factors (16).

The RKR-IPA program set yielded results that agreed very well with similar work in the literature (5). However, interest does exist in finding a more efficient program set based on some other technique than RKR. This technique should avoid the following shortcomings of the RKR program set. First, the RKR-IPA programs require considerable computer resources and are therefore expensive to use. These programs can not run on just any minicomputer that may be available. They require full mainframe support. Second, researchers have observed anomalous behavior of the RKR potential (16), (19), (23), (25), (29). The curve sometimes bends over, or decreases rapidly for higher energy states. These effects are usually attributed to an incorrect set of molecular constants. Wells, Smith, and Zare concluded that the RKR method is very sensitive to errors or inconsistencies in the experimental data (27). This has sparked interest in a technique that uses the experimental data directly to find the potential that yields the best fit to the measured energy differences.

Previous work to find a new technique or better version of RKR have not made any significant improvements over RKR,

Potential Energy Models

Some knowledge of the potential energy of diatomic molecules is required before the wave equation can be solved for the energy eigenvalues or wave functions. The exact form of a function describing the potential energy is not in general known. However, in many cases the potential appears to have a form illustrated by Figure II-4. The two constants shown are the molecule's dissociation energy D_e and the equilibrium distance between nuclei r_e.

One of the most widely used potential energy models is the Morse function (10:101):

$$V_{(r)} = D_e\left[1 - e^{-\beta(r-r_e)}\right]^2 \tag{16}$$

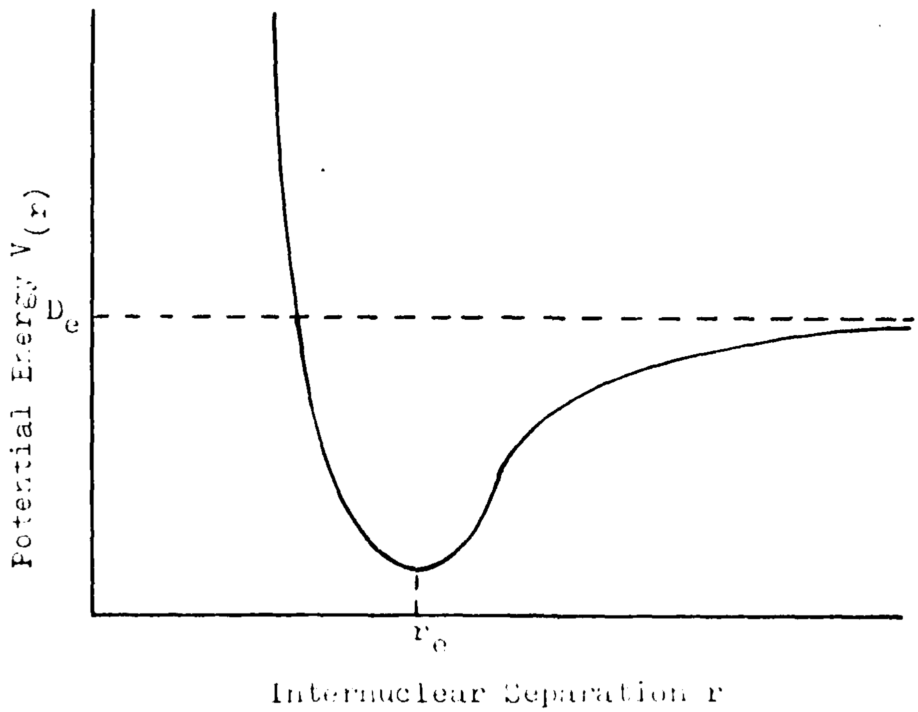


Fig. II-4. A Potential Energy Curve of a Diatomic Molecule

Now the vector $\underline{y}$ can be back-transformed into vector $\underline{x}$, which contains the values of the energy levels, by solving $\underline{L}^T \underline{x} = \underline{y}$. $L^T_{pp} = 0$ since $\underline{L}$ and $\underline{L}^T$ have the same diagonal. This means that x_p is arbitrary since $0 \; x_p = y_p$ where y_p is also arbitrary. The vector element x_p is chosen to be zero since it affects the values of other vector elements x_i $(i < p)$. The remaining elements of $\underline{y}$ are related to $\underline{x}$ by

$$y_i = L^T_{ii} x_i + \sum_{k=i+1}^{p-1} L^T_{ik} x_k \quad (i=1,2,\ldots,(p-1)) \tag{14}$$

since $\underline{L}^T$ is upper triangular. Inverting and remembering $L^T_{ik} = L_{ki}$ Eq (14) becomes:

$$x_i = \frac{y_i - \sum_{k=i+1}^{p-1} L_{ki} x_k}{L_{ii}} \quad (i=(p-1),(p-2),\ldots,1) \tag{15}$$

The value of x_{p-1} is computed first since it is needed to compute the value of x_{p-2}.

The vector $\underline{x}$ now contains the energy levels that best fit the spectroscopic data. However, the highest energy level ϵ_p has a value of zero and all other levels are negative. The energy levels may now be shifted so that ϵ_1 is either zero, or the value the Dunham equation (Eq (1)) yields.

This least squares method yields values for only the energy levels represented in the data. Missing values may be found from the Dunham equation (Eq (1)).

$$L_{ij} = \frac{A_{ij} - \sum_{k=1}^{(j-1)} L_{ik}L_{jk}}{L_{jj}}$$

$$L_{jj} = (A_{jj} - \sum_{k=1}^{(j-1)} L_{jk}^2)^{\frac{1}{2}} \tag{10}$$

where $L_{ij} = 0$ if $j > i$ since $\underline{L}$ is lower triangular. Also, the last diagonal element L_{pp} is zero since $\underline{A}$ is singular.

The solution of $\underline{L}\ \underline{y} = \underline{b}$ is found by inverting

$$b_i = \sum_{k=1}^{(i-1)} L_{ik}y_k + L_{ii}y_i \tag{11}$$

to get

$$y_i = \frac{b_i - \sum_{k=1}^{(i-1)} L_{ik}y_k}{L_{ii}} \tag{12}$$

This method yields only $p - 1$ values for $\underline{y}$ since $L_{pp} = 0$ and Eq (12) can not be solved for y_p. This means that y_p is arbitrary and can be chosen to be zero. However, the condition

$$b_p = \sum_{k=1}^{(p-1)} L_{pk}y_k \tag{13}$$

must be met if the problem is to be consistent.

level difference information. The solution is unique only in terms of differences $\epsilon_j - \epsilon_i$, not in the value ϵ_i acquires.

Since $\underline{A}$ is positive semi-definite (28:28-30), the Cholesky decomposition (28:229) (4:Sec 8 1-12) can be used to state the problem in a solvable form. First, a lower triangular matrix $\underline{L}$ is found so that $\underline{A} = \underline{L}\ \underline{L}^T$ ($\underline{L}^T$ is the transpose of $\underline{L}$). Then the problem becomes $\underline{L}\ \underline{y} = \underline{b}$ where $\underline{y} = \underline{L}^T\underline{x}$.

The decomposition of $\underline{A}$ occurs as follows. If $\underline{A} = \underline{L}\ \underline{L}^T$, then element A_{ij} of $\underline{A}$ is:

$$A_{ij} = \sum_{k=1}^{\min(i,j)} L_{ik}L^T_{kj} = \sum_{k=1}^{\min(i,j)} L_{ik}L_{jk} \tag{8}$$

where $L^T{}_{kj} = L_{jk}$ since $\underline{L}^T$ is the transpose of $\underline{L}$. Three cases arise when evaluating Eq (8):

$$\begin{aligned}
i=j \quad & A_{jj} = L^2_{jj} + \sum_{k=1}^{(j-1)} L^2_{jk} \\
i>j \quad & A_{ij} = L_{ij}L_{jj} + \sum_{k=1}^{(j-1)} L_{ik}L_{jk} \\
i<j \quad & A_{ij} = L_{ij}L_{ii} + \sum_{k=1}^{(i-1)} L_{ik}L_{jk}
\end{aligned} \tag{9}$$

Inverting Eqs (9) yields

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} n=1 & & n=j & & n=k & & \end{array} \\
\begin{array}{c} m=1 \\ \\ \\ m=k \\ \\ \end{array} &
\begin{bmatrix}
\sum_{i=1}^{p}(w_{i1}+w_{1i}) & \cdots & -(w_{1j}+w_{j1}) & \cdots & -(w_{1k}+w_{k1}) & \cdots \\
\vdots & \ddots & \vdots & & \vdots & \\
-(w_{k1}+w_{1k}) & \cdots & -(w_{kj}+w_{jk}) & \cdots & \sum_{i=1}^{p}(w_{ik}+w_{ki}) & \cdots \\
\vdots & & \vdots & \ddots & \vdots &
\end{bmatrix}
\begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_k \\ \vdots \end{bmatrix}
=
\begin{bmatrix} \sum_{j=1}^{p}(w_{1j}l_{1j}-w_{j1}l_{j1}) \\ \vdots \\ \sum_{j=1}^{p}(w_{kj}l_{kj}-w_{jk}l_{jk}) \\ \vdots \end{bmatrix}
\end{array}
$$

$\underline{\underline{W}}$ an mxn matrix m=n=p $\qquad$ $\underline{x}$ a px1 matrix $\quad = \quad$ $\underline{b}$ a px1 matrix

Fig. II-3. The Least Squares Problem in Matrix Form

$$0 = \sum_{i=1}^{p} w_{ik}(l_{ik}-\epsilon_i+\epsilon_k) - \sum_{j=1}^{p} w_{kj}(l_{kj}-\epsilon_k+\epsilon_j) \qquad (6)$$

A final rearrangement of terms yields:

$$\epsilon_k \sum_{i=1}^{p}(w_{ik}+w_{ki}) - \sum_{j=1}^{p}(w_{kj}+w_{jk})\epsilon_j = \sum_{j=1}^{p}(w_{kj}l_{kj}-w_{jk}l_{jk}) \qquad (7)$$

The first term of Eq (7) is the sum of the weighting factors for all transitions ending at level ϵ_k plus the sum of the weighting factors for all transitions beginning at level ϵ_k times the value of ϵ_k. The second term is the sum over j of the weighting factor for the transitions between levels ϵ_k and ϵ_j times the value of ϵ_j. Finally, the third term of Eq (7) is the sum over j of the weighted transition from level ϵ_k to level ϵ_j minus the weighted transition from level ϵ_j to level ϵ_k. Only observed transitions are used in Eq (7).

This relationship can be more conveniently solved by rewriting Eq (7) in matrix form as shown in Figure II-3. This is the common linear problem $\underline{A}\underline{x} = \underline{b}$ where $\underline{A}$ is a p by p square matrix, and the vectors $\underline{x}$ and $\underline{b}$ are p by 1 matricies. The diagonal elements of $\underline{A}$ (A_{mn} m = n) are the sum of (w_{ik} + w_{ki}) i = 1 to p. The off diagonal elements (A_{mn} m $\neq$ n) are $-(w_{mn} + w_{nm})$.

The problem is complicated by the fact that $\underline{A}$ is singular, possessing no inverse $\underline{A}^{-1}$ such that $\underline{x} = \underline{A}^{-1}\,\underline{b}$. $\underline{A}$ is singular since the spectroscopic data contain only energy

inconsistencies in the additional data. The following analysis finds the set of energy level values that best resolves the inconsistencies of the entire set of data.

The weighted linear least squares technique finds a set of energy levels with the smallest sum S for the p observed levels.

$$S = \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} w_{ij}(l_{ij} - \epsilon_i + \epsilon_j)^2 \tag{2}$$

The weighting factor w_{ij} allows more accurately determined data to make a larger contribution to the sum S. The minimum of the sum S occurs when its derivatives with respect to the energy levels of interest are zero:

$$\frac{\partial S}{\partial \epsilon_k} = 0 \tag{3}$$

The derivative is

$$0 = \sum_{i=1}^{p} \sum_{j=1}^{p} w_{ij}(l_{ij} - \epsilon_i + \epsilon_j)(\delta_{jk} - \delta_{ik}) \tag{4}$$

since

$$\frac{\partial \epsilon_m}{\partial \epsilon_n} = \delta_{mn} = \begin{cases} 1 \text{ if } m=n \\ 0 \text{ if } m \neq n \end{cases} \tag{5}$$

Using this property (Eq (5)) of the Kronecker delta δ_{mn}, Eq (4) becomes:

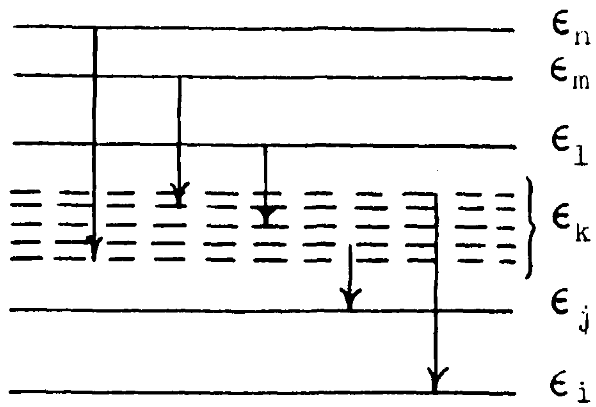


Fig. II-2. Inconsistencies of Spectroscopic Data

transition lines involving the same energy level. This is illustrated in Figure II-2. The observed differences between level ϵ_k and the other levels $\epsilon_{i,j,l,m,n}$ (lines $l_{k,\alpha}$ α = i,j,l,m,n) do not agree with each other. Each line supposes that the level ϵ_k is at a different position than the other lines. If level ϵ_k were to be pinned down to a definite value, then all the other levels would shift. This is not a problem in this simple case. However, a problem arises when other transitions are observed not involving level ϵ_k. This additional data will most likely be inconsistent as the data involving level ϵ_k were. Now the shifting of energy levels caused by pinning ϵ_k to a definte value further compound the

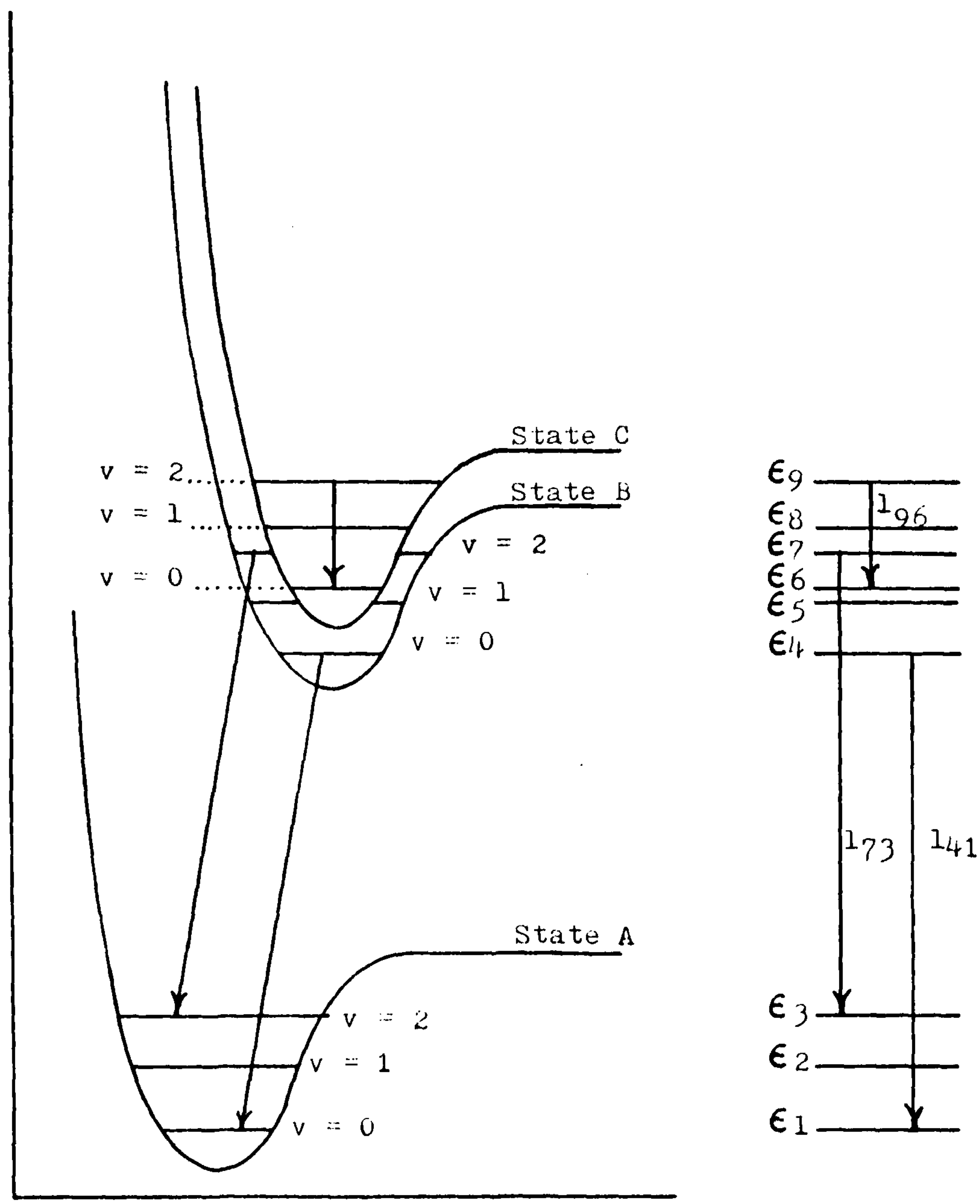


Fig. II-1. Electronic Transition Lines

coefficients are often published in place of the data they represent. The Dunham coefficient Y_{ij} is a coefficient of an infinite power series of vibrational and rotational quantum numbers (v and J). Given enough terms, this series accurately represents the energy T(v,J) of the quantum state represented by v and J (6:725). The Dunham equation is:

$$T_{(v,J)} = \sum_{ij} Y_{ij}(v+\tfrac{1}{2})^{i}J^{j}(J+1)^{j} \tag{1}$$

Pow describes a good method of determining the Dunham coefficients from spectroscopic data (16:4-8).

Alternatively, the researcher may use the spectroscopic data directly to determine the observed energy levels by a weighted least squares fit.

First, a set of transition lines l_{ij} are observed and assigned. The assignment identifies the transition's initial and final electronic and vibrational states. Each transition is from an initial energy level ϵ_i to a final level ϵ_j. These levels may or may not belong to the same electronic state as shown in Figure II-1. For example, line l_{73} represents a transition from electronic state B, v = 2 to state A, v = 2. Also, line l_{96} represents the transition from state C, v = 2 to state C, v = 0.

Then, the least squares technique is used to resolve inconsistencies between the data. A transition line involving one energy level may not agree with other

II. Literature Background and Theory

This section develops concepts and theory required to understand the programs presented later. The goal of this effort is the calculation of the transition probability (Franck-Condon factor) between two energy states of a diatomic molecule. The wave functions describing these states are used to calculate the Franck-Condon factor, and are found by solving the Schrodinger wave equation. However, knowledge of the molecule's kinetic and potential energy is required before solving for the wave functions. Therefore, a discussion is first presented on determining the appropriate electronic and vibrational energy levels of the system. Then, two potential energy models are presented. This section is followed by a detailed outline of a finite element solution of the wave equation. Finally, the calculation of Franck-Condon factors is discussed (21).

Determination of Energy Levels

There are two starting points for determining the energy levels of a diatomic molecular system. The researcher may either use Dunham coefficients to calculate the energy levels, or they may be derived directly from spectral data.

Dunham coefficients are combinations of molecular constants derived from spectroscopic data. These

not be important, as long as it allows the flexibility to modify the repulsive and attractive branches independently.

This program was validated against the analytic solutions of the simple harmonic oscillator. Two programs have been written to aid the researcher in defining the energy levels used as input. One program uses a least squares technique to find the set of energy levels that best fit a set of spectroscopic, transition data. The second uses Dunham coefficients to calculate approximate values for missing energy levels. The last program written computes a 25 by 25 (v' = 0 to 24; v" = 0 to 24) table of Franck-Condon factors between two electronic states.

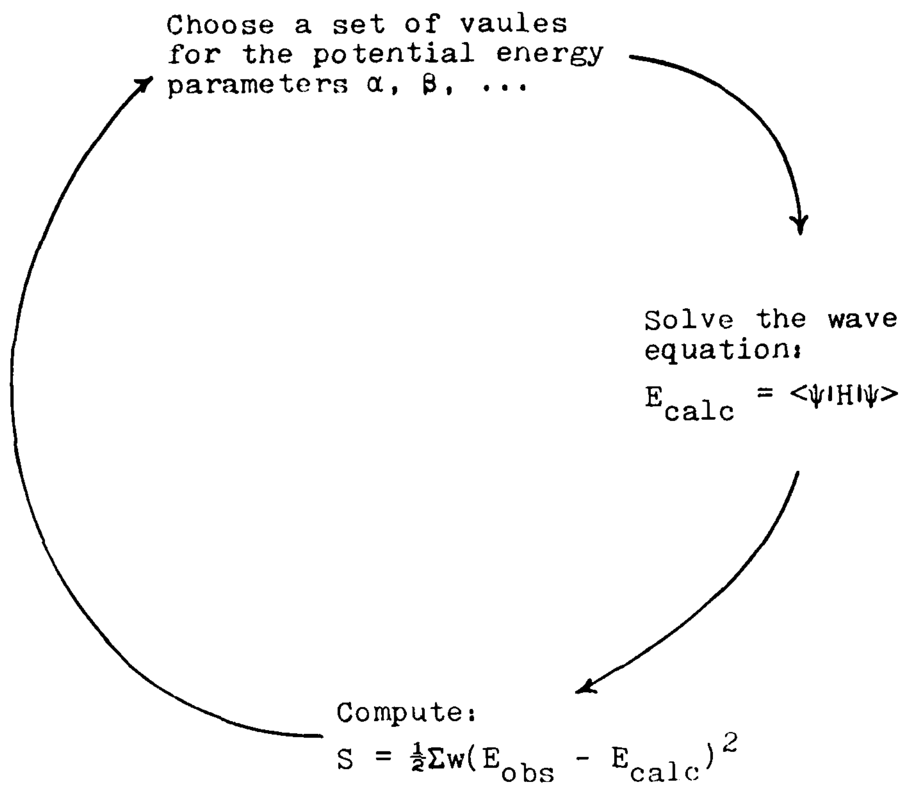


Fig. 1-1. Minimization of the Least Squares Sum

or are still computationally expensive (8), (12), (24).

Objective

The objective of this work is to develop and validate a program set based on a quantum mechanical approach. These programs should be transportable to minicomputers such as a VAX 11/780, HP 1000, Harris 800 and be inexpensive to use. Also, they should use observed energy levels as input, not a set of molecular constants derived from these observed levels.

Approach

A program has been written which solves the wave equation using the finite element method. This program uses a potential energy function defined in terms of parameters the program can change. A non-linear minimization routine is used to find the set of parameters which characterize the potential that best fits the observed energy levels in a least squares sense. This approach is illustrated in Figure I-1. The minimization routine selects a set of parameters, solves the wave equation, and computes a sum of the difference between observed and calculated energy levels squared. This process is repeated until the set of parameters is found that yields the smallest energy difference sum.

This approach allows the researcher to fit a potential energy model to his data. The specific model chosen should

The molecular constant β is defined as (10:101):

$$\beta = \left(\frac{2\pi c\mu}{D_e h}\right)^{\frac{1}{2}}\omega_e = 1.2177\omega_e\left(\frac{\mu}{D_e}\right) \qquad (17)$$

where μ is the reduced mass of the system and ω_e is a vibrational frequency. Tellinghuisen and Henderson observed that the Morse function seems to accurately describe the repulsive branch of the potential (24).

Another potential energy model is the Lennard-Jones potential:

$$V_{(r)} = D_e\left(\left(\frac{r_e}{r}\right)^{12} - 2\left(\frac{r_e}{r}\right)^6\right) \qquad (18)$$

This potential is considered since it is the sum of a repulsive and attractive potential. Therefore, each branch may be modified independently of the other branch.

Whichever model is chosen, the potential is described in terms of parameters than can be altered before solving the wave equation. For example, the powers 12 and 6 of Eq (18) might be replaced by α and β. Then α and β are each allowed to have a certain value, say $\alpha = 12$ and $\beta = 6$ or $\alpha = 11.39$ and $\beta = 5.06$. The method then is to vary the potential parameters and solve the wave equation in an iterative fashion. Given enough iterations the set of parameter values will be found for the potential energy model making the best fit to the observed energy levels.

Care must be exercised when the potential energy function is parameterized to ensure that the model is still an accurate description of the molecule. For example, if the Lennard-Jones potential were parameterized as

$$V_{(r)} = D_e\left(\left(\frac{r_e}{r}\right)^{\alpha} - 2\left(\frac{r_e}{r}\right)^{\beta}\right) \tag{19}$$

one would find that the function's minimum no longer occurs at r_e if $\alpha = 11.39$ and $\beta = 5.06$. However, if the potential were parameterized as

$$V_{(r)} = D_e \frac{\beta}{\alpha-\beta}\left(\left(\frac{r_e}{r}\right)^{\alpha} - \frac{\alpha}{\beta}\left(\frac{r_e}{r}\right)^{\beta}\right) \tag{20}$$

then the function behaves well for all α and β. The function described by Eq (20) is the Mie potential (14:311). Moelwyn-Hughes points out that both the Morse and Lennard-Jones functions are special forms of the Mie function (14:311-315).

Numerical Solution of the Wave Equation

The wave functions used to calculate Franck-Condon factors are derived by solving the Schrodinger wave equation. The procedure uses the finite element method to solve for the energy eigenvalues of a particular set of potential energy parameters. The calculated energy eigenvalues are compared with observed energy levels in a weighted least squares sense. Then, a new set of potential energy parameters is

generated, the wave equation solved, and energy levels compared. This process is repeated until the parameter set that best fits the measured energy levels is found. Then the eigenvectors are computed and normalized.

The expected values of the energy operator are:

$$E = \text{ext}\left\{\frac{\int dx \psi^{+} H \psi}{\int dx \psi^{+} \psi}\right\} \tag{21}$$

where ext { } means the extremum of the backeted quantity. The extrema of a function are its minima, maxima, and saddle points. Assuming spherical symmetry, Eq (21) becomes

$$E = \text{ext}\left\{\frac{4\pi\int_0^\infty dr r^2 \psi^{+}_{(r)} \left(\frac{-\hbar^2}{2\mu}\nabla^2 + V_{(r)}\right) \Psi_{(r)}}{4\pi\int_0^\infty dr r^2 \psi^{+}_{(r)} \Psi_{(r)}}\right\} \tag{22}$$

where the Laplacian operator is:

$$\nabla^2 = \frac{1}{r^2} \frac{d}{dr} r^2 \frac{d}{dr} \tag{23}$$

The wave function $\psi_{(r)}$ is replaced by a function $U_{(r)}$ so that:

$$\Psi_{(r)} = \frac{U_{(r)}}{r} = \frac{U^{+}_{(r)}}{r} \tag{24}$$

(See French (7:199-201) for another discussion on this substitution). Then Eq (22) becomes:

$$E = \text{ext}\left\{\frac{\int_0^\infty dr\, r U_{(r)} \left(\frac{-\hbar^2}{2\mu} \frac{1}{r^2} \frac{d}{dr} r^2 \frac{d}{dr} + V_{(r)}\right) \frac{U_{(r)}}{r}}{\int_0^\infty dr\, U^2_{(r)}}\right\} \tag{25}$$

Since

$$\begin{aligned} r^2 \frac{d}{dr}\left(\frac{U_{(r)}}{r}\right) &= r^2\left(\frac{-U_{(r)}}{r^2} + \frac{U'_{(r)}}{r}\right) \\ &= -U_{(r)} + rU'_{(r)} \end{aligned} \tag{26}$$

the kinetic energy term of Eq (25) is rewritten as:

$$\begin{aligned} \frac{1}{r^2} \frac{d}{dr}\left[r^2 \frac{d}{dr}\left(\frac{U_{(r)}}{r}\right)\right] &= \frac{1}{r^2} \frac{d}{dr}\left(-U_{(r)} + rU'_{(r)}\right) \\ &= \frac{1}{r^2}\left(-U'_{(r)} + U'_{(r)} + rU''_{(r)}\right) \\ &= \frac{U''_{(r)}}{r} \end{aligned} \tag{27}$$

Therefore Eq (25) is now

$$E = \text{ext}\left\{\frac{\int_0^\infty dr\, U_{(r)} \left(\frac{-\hbar^2}{2\mu} \frac{d^2}{dr^2} + V_{(r)}\right) U_{(r)}}{\int_0^\infty dr\, U^2_{(r)}}\right\} \tag{28}$$

When integrated by parts, Eq (28) becomes

$$E = \mathrm{ext}\left\{\frac{\int_0^\infty dr\left[\frac{\hbar^2}{2\mu}\left(\frac{dU_{(r)}}{dr}\right)^2 + U_{(r)}^2 V_{(r)}\right]}{\int_0^\infty dr U_{(r)}^2}\right\} \tag{29}$$

The finite element method can now be used to solve Eq (29). First a uniform grid is overlayed on the wave function as indicated in Figure II-5. Each grid point r_α is a node. The natural coordinate system (Fig II-6) (3:88) simplifies the problem. Each position r is described in terms of the local grid boundaries r_i and r_{i+1}, and the natural coordinates l_1 and l_2 as:

$$r = r_i + l_2 h = r_{i+1} - l_1 h \tag{30}$$

where h is the grid element size

$$h = r_{i+1} - r_i \tag{31}$$

and $l_1 + l_2 = 1$.

The approximate solutions of Eq (29) found by the finite element method will converge to the true solution as the grid element size is made smaller. However, the interpolating polynomial chosen to approximate $U_{(r)}$ must satisfy the requirements of completeness and compatibility presented by Rao (17:114-115). These requirements are met if a basis set of terms cubic in l_1 and l_2 are used to approximate $U_{(r)}$.

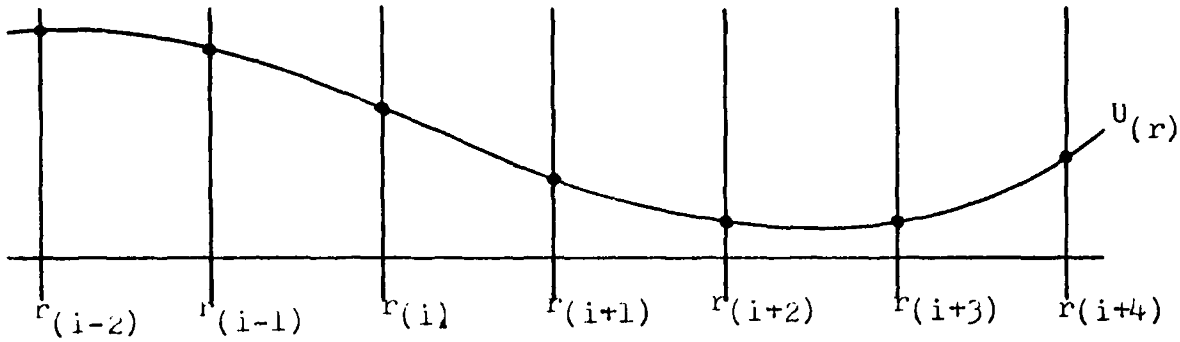


Fig. II-5. Finite Element Grid

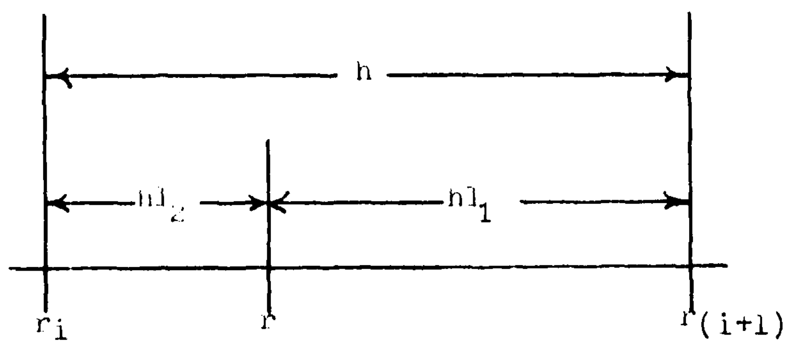


Fig. II-6. Natural Coordinate System

Then the requirement is that the approximating function and its first derivatives are continuous at each node r_i. The function $U_{(r)}$ is approximated by:

$$U_{(r)} = U_{(l_1,l_2)} \qquad r_i \le r \le r_{i+1}$$
$$U_{(l_1,l_2)} = U_0 f_{1(l_1,l_2)} + U_0' f_{2(l_1,l_2)} + U_1 f_{3(l_1,l_2)} + U_1' f_{4(l_1,l_2)} \tag{32}$$

where U_0 = value of $U_{(r)}$ at r_i
U_0' = slope of $U_{(r)}$ at r_i
U_1 = value of $U_{(r)}$ at r_{i+1}
U_1' = slope of $U_{(r)}$ at r_{i+1}

Then the boundary conditions at the left edge of the grid element ($l_1 = 1$, $l_2 = 0$) are:

$$\begin{array}{ll} f_{1(1,0)} = 1 & f_{1(1,0)}' = 0 \\ f_{2(1,0)} = 0 & f_{2(1,0)}' = 1 \\ f_{3(1,0)} = 0 & f_{3(1,0)}' = 0 \\ f_{4(1,0)} = 0 & f_{4(1,0)}' = 0 \end{array} \tag{33}$$

The boundary conditions at the right edge ($l_1 = 0$, $l_2 = 1$) are:

$$\begin{array}{ll} f_{1(0,1)} = 0 & f_{1(0,1)}' = 0 \\ f_{2(0,1)} = 0 & f_{2(0,1)}' = 0 \\ f_{3(0,1)} = 1 & f_{3(0,1)}' = 0 \\ f_{4(0,1)} = 0 & f_{4(0,1)}' = 1 \end{array} \tag{34}$$

The function $U_{(r)}$ should be zero and have zero slope at $r = 0$ and $r = \infty$. This condition is not enforced, but should be a result of this analysis as the function gets far from the potential.

The basis set for cubics in l_1 and l_2 is $\{l_1{}^3;\ l_1{}^2 l_2;\ l_1 l_2{}^2;\ l_2{}^3\}$. The first derivatives of the basis set are $\{-3l_1{}^2/h;\ (-2l_1 l_2 + l_1{}^2)/h;\ (-l_2{}^2 + 2l_1 l_2)/h;\ 3l_2{}^2/h\}$ since

$$\frac{d\gamma}{dr} = \frac{d\gamma}{dl_1}\frac{dl_1}{dr} + \frac{d\gamma}{dl_2}\frac{dl_2}{dr} \tag{35}$$

and

$$\frac{dl_1}{dr} = -\frac{1}{h}$$
$$\frac{dl_2}{dr} = \frac{1}{h} \tag{36}$$

so that Eq (35) becomes

$$\frac{d\gamma}{dr} = \frac{1}{h}\left(\frac{d\gamma}{dl_2} - \frac{d\gamma}{dl_1}\right) \tag{37}$$

The combination of basis functions that satisfy the boundary conditions (Eqs (33) and (34)) is:

$$f_1 = l_1^3+3l_1^2l_2 \qquad f_1' = -(6l_1l_2)/h$$

$$f_2 = hl_1^2l_2 \qquad f_2' = (l_1^2-2l_1l_2)/h$$

$$f_3 = l_2^3+3l_1l_2^2 \qquad f_3' = (6l_1l_2)/h \tag{38}$$

$$f_4 = -hl_1l_2^2 \qquad f_4' = (l_2^2-2l_1l_2)/h$$

The function $U_{(r)}$ and its first derivative are now approximated by:

$$\begin{aligned} U_{(l_1,l_2)} &= U_0(l_1^3+3l_1^2l_2)+U_0'(hl_1^2l_2)+U_1(l_2^3+l_1l_2^2) \\ &\qquad +U_1'(-hl_1l_2^2) \\ &= U_0l_1^3+(3U_0+U_0'h)l_1^2l_2+(3U_1-U_1'h)l_1l_2^2+U_1l_2^3 \end{aligned} \tag{39}$$

and

$$\begin{aligned} \frac{dU_{(l_1,l_2)}}{dr} &= U_0\left(\frac{-6l_1l_2}{h}\right)+U_0'(-2l_1l_2+l_1^2)+U_1\left(\frac{6l_1l_2}{h}\right) \\ &\qquad +U_1'(-2l_1l_2+l_2^2) \\ &= U_0'l_1^2+(-3U_0/h-U_0'+3U_1/h-U_1')2l_1l_2+U_1'l_2^2 \end{aligned} \tag{40}$$

The integrals of Eq (29) may now be rewritten as

$$I_1 = \frac{\hbar^2}{2\mu}\int_0^{h} dl_1\,dl_2\left[U_0'l_1^2+\left(\frac{3U_0}{h}-U_0'+\frac{3U_1}{h}-U_1'\right)2l_1l_2+U_1'l_2^2\right]^2 \tag{41}$$

$$\begin{aligned} I_2 = \int_0^h dl_1\,dl_2[&U_0l_1^3+(3U_0+U_0'h)l_1^2l_2+(3U_1-U_1'h)l_1l_2^2 \\ &+U_1l_2^3]^2V_{(l_1,l_2)} \end{aligned} \tag{42}$$

$$I_3=\int_0^h dl_1\,dl_2\left[U_0 l_1^3+(3U_0+U_0'h)l_1^2 l_2+(3U_1-U_1'h)l_1 l_2^2+U_1 l_2^3\right]^2 \tag{43}$$

for each grid element where

$$I_1+I_2=\int_0^\infty dr\left[\frac{\hbar^2}{2\mu}\left(\frac{dU_{(r)}}{dr}\right)^2+U_{(r)}^2 V_{(r)}\right]$$

$$I_3=\int_0^\infty dr\,U_{(r)}{}^2 \tag{44}$$

These integrals are solvable by the simple relation (3:312):

$$\int_0^h dl_1\,dl_2\,l_1^p l_2^q=\frac{h\,p!\,q!}{(p+q+1)!} \tag{45}$$

Since U_o, U_o', U_1, and U_1' are constants with respect to the coordinates l_1 and l_2, the result is the integral of a polynomial in terms of l_1 and l_2. Then, if each integral is separated into a sum of integrals and constants are taken out of the integral, Eq (45) solves each term of the polynomial. The integral I_2 (Eq (42)) is complicated at this point by the potential energy term $V(l_1,l_2)$ and will be discussed next. When solved, I_1 (Eq (41)) and I_3 (Eq (43)) result in a polynomial in the boundary values U_o, U_o', U_1, and U_1' of the grid element. For example, I_3 becomes

$$I_3 = \frac{1872hU_0^2}{7!} + \frac{264h^2U_0U_0'}{7!} + \frac{648hU_0U_1}{7!} + \ldots \quad (46)$$

This expression is more convenient when written in matrix form as:

$$I_3 = \begin{bmatrix} U_0 & U_0' & U_1 & U_1' \end{bmatrix} \frac{h}{7!} \begin{bmatrix} 1872 & 264h & 648 & -156h \\ 264h & 48h^2 & 156h & -36h^2 \\ 648 & 156h & 1872 & -264h \\ -156h & -36h^2 & -264h & 48h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_0' \\ U_1 \\ U_1' \end{bmatrix} \quad (47)$$

Similarly, the integral I_1 becomes

$$I_1 = \begin{bmatrix} U_0 & U_0' & U_1 & U_1' \end{bmatrix} \frac{\hbar^2}{240\mu h} \begin{bmatrix} 144 & 12h & -144 & 12h \\ 12h & 16h^2 & -12h & -4h^2 \\ -144 & -12h & 144 & -12h \\ 12h & -4h^2 & -12h & 16h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_0' \\ U_1 \\ U_1' \end{bmatrix} \quad (48)$$

The integral over the entire grid is then reduced to a matrix problem composed of one matrix I_i for each grid element i. For example, the overlap integral becomes:

$$I_{III} = \int_a^b drU_{(r)}^2 = \underline{z}^T\underline{S}\ \underline{z} \quad (49)$$

where a and b are the first and last nodes of the grid. Then $\underline{z}^T\underline{S}\ \underline{z}$ is constructed as illustrated in Figure II-7 where the i^{th} sub-matricies of $\underline{z}$ and $\underline{S}$ are shown in Figure II-8. Each

The value and slope of $U_{(r)}$ at the 1^{st} grid point

$$\underline{z}^T\underline{\underline{S}}\ \underline{z} = \begin{bmatrix} U_1 & U_1' & U_2 & U_2' & U_3 & U_3' & U_4 & U_4' & \cdots \end{bmatrix} \begin{bmatrix} S_1 & S_1 & S_1 & S_1 & & & \\ S_1 & S_1 & S_1 & S_1 & & & \\ S_1 & S_1 & S_1+S_2 & S_1+S_2 & S_2 & S_2 & \\ S_1 & S_1 & S_1+S_2 & S_1+S_2 & S_2 & S_2 & \\ & & S_2 & S_2 & S_2+S_3 & S_2+S_3 & \\ & & S_2 & S_2 & \cdot & & \\ & & & & & \cdot & \\ & & & & & & \cdot \end{bmatrix} \begin{bmatrix} U_1 \\ U_1' \\ U_2 \\ U_2' \\ U_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Sub-Matrix $\underline{S}_1$

Sub-Matrix $\underline{S}_2$

Fig. II-7. The Matrix Problem $\underline{z}^T\underline{S}\ \underline{z}$

$$\underline{S}_i = \begin{bmatrix} \frac{1872h}{7!} & \frac{264h^2}{7!} & X & X \\ \frac{264h^2}{7!} & X & X & X \\ X & X & X & X \\ X & X & X & X \end{bmatrix}$$

Fig. II-8. The Sub-Matricies of $\underline{z}^T\underline{S}\ \underline{z}$

sub-matrix corresponds to one grid element. The sub-matricies $\underline{z}^i$ and $\underline{S}^i$ overlap with sub-matricies $\underline{z}^{i-1}$, $\underline{z}^{i+1}$ and $\underline{S}^{i-1}$, $\underline{S}^{i+1}$ since the $(i-1)^{th}$ and i^{th} grid elements share the same grid boundaries, as do the $(i+1)^{th}$ and i^{th} elements:

$$\begin{aligned} U_1^{i-1} &= U_0^i \\ U_1'^{\,i-1} &= U_0'^{\,i} \\ U_1^i &= U_0^{i+1} \\ U_1'^{\,i} &= U_0'^{\,i+1} \end{aligned} \qquad (50)$$

When element S_{mn} of $\underline{S}$ or z_m of $\underline{z}$ correspond to the overlap of sub-matricies $\underline{S}^i$, $\underline{S}^{i+1}$ or $\underline{z}^i$, $\underline{z}^{i+1}$ the value of S_{mn} or z_m is the sum of the corresponding values of the sub-matricies.

The integral I_2 (Eq(42)) is solved in the same manner; however, the potential energy function $V_{(r)}$ must first be approximated by $V(l_1,l_2)$. The potential $V_{(r)}$ is written as a combination of the same cubic basis set used for the function $U_{(r)}$. This keeps the same level of accuracy between the approximating functions $U(l_1,l_2)$ and $V(l_1,l_2)$. Specifically, $V_{(r)}$ becomes:

$$V_{(r)}=V_{(l_1,l_2)} \qquad r_i \leq r \leq r_{i+1}$$
$$V_{(l_1,l_2)}=V_0 f_{1(l_1,l_2)}+V_0' f_{2(l_1,l_2)}+V_1 f_{3(l_1,l_2)}+V_1' f_{4(l_1,l_2)} \tag{51}$$

where the f_α's are the same functions used earlier (Eq (38)).

Then the integral I_2 (Eq (42)) for the i^{th} grid element is

$$I_2=\int_{r_i}^{r_{i+1}} dr U^*_{(r)} V_{(r)}$$
$$=\int_0^h dl_1 dl_2 [U_0 l_1^3+(3U_0+U_0'h)l_1^2 l_2+(3U_1-U_1'h)l_1 l_2^2+U_1 l_2^3]^2 [V_0 l_1^3 \tag{52}$$
$$+(3V_0+V_0'h)l_1^2 l_2+V_1 l_2^3+(3V_1-V_1'h)l_1 l_2^2]$$

so

$$I_2 = I_{21} + I_{22} + I_{23} + I_{24} \tag{53}$$

After the same manipulations described before, the $I_{2\alpha}$'s are evaluated as:

File COMP

```
MS
GE SPOCK
JE,2150 !BBBA
!BBBA CO VLCN2 SPOCK
ED SPOCK
TA
E,3
C,6-8, &2
$JE,170,!NED
!NED UP
MO $A=ON
FR 5,6
$JE,373,!NEXT
!NEXT *SAUF77.LPVW&O &1
$JE !F77ERR
$PR compiled ok...
$JU !ELOLD
!F77ERR $PR Ooops...
!ELOLD EL &2
$JE,312,!ELZZZ
$JE,2170,!NEXT2
$JU !NEXT2
!ELZZZ EL ZZZ
$JE,2170,!NEXT2
!NEXT2 JS SPOCK
MO $A=OFF
EL SPOCK
AS 6=*
$PR done and did.
ME
```

File VLCN2

```
$PR vulcanization...
VU.B
NAME ZZZ
LI,2039EFPH*BINDER,1000AFIT*IMSLIB,*SAUL77,*LIBERY
BE
```

Fig. III-1. Fortran 77 Compile and Link Macro

The job control statements required to compile and link these programs is given in Figure III-1. These statements compose a macro. A macro is a file of commands referenced by the file name in which the statements are stored. For example, if the statements of Fig III-1 are stored in file COMP, the macro is executed by COMP DIATOM DIA or COMP.D DIATOM DIA. In the first example, the program stored in the file DIATOM (Appendix F) is compiled and linked. The executable module is stored in the file DIA. In the second example DIATOM is compiled using the debug (D) compiler option. The source code in the file DIATOM may either contain all user routines, or references to other files containing more source code. The $ADD facility causes another file (filename = xxxx) to be inserted where $ADD xxxx occurs (for examples, see Appendices B, D, F and H). The executable code is run by entering the name of the file containing it.

Before executing the program, the user must attach all input/output files. If the program reads from logical unit 11 and writes to logical unit 13, then they must be attached to the files by the AS (assign) command (e.g. AS 11 = INFN and AS 13 = OUTFN). INFN and OUTFN refer to the file names involved. The macro shown in Figure III-2 makes all necessary assignments and executes the program DIATOM stored in executable form in the file DIA.

III. Computer Programs

This section presents the computer code written to implement the theory outlined in Chapter II. The programs developed are called DUNHAM, EFIT, DIATOM, and FCFACT. The environment these programs run in is first discussed. This includes the compile, link, and execution steps. Then each individual program is discussed, including its purpose, cababilities, and major sub-program tasks. Finally, the interrelationships between the programs is presented. Program flow diagrams and source code are in the Appendicies A - H. Appendix I is a legend for the flow diagrams.

Program Environment

The programs run on AFIT's Harris 800 minicomputer under the VOS operating system. Each program may run in either a batch or interactive-terminal environment. The Harris is a 24 bit machine with two words used to represent a real variable. Single precision mathematics is used.

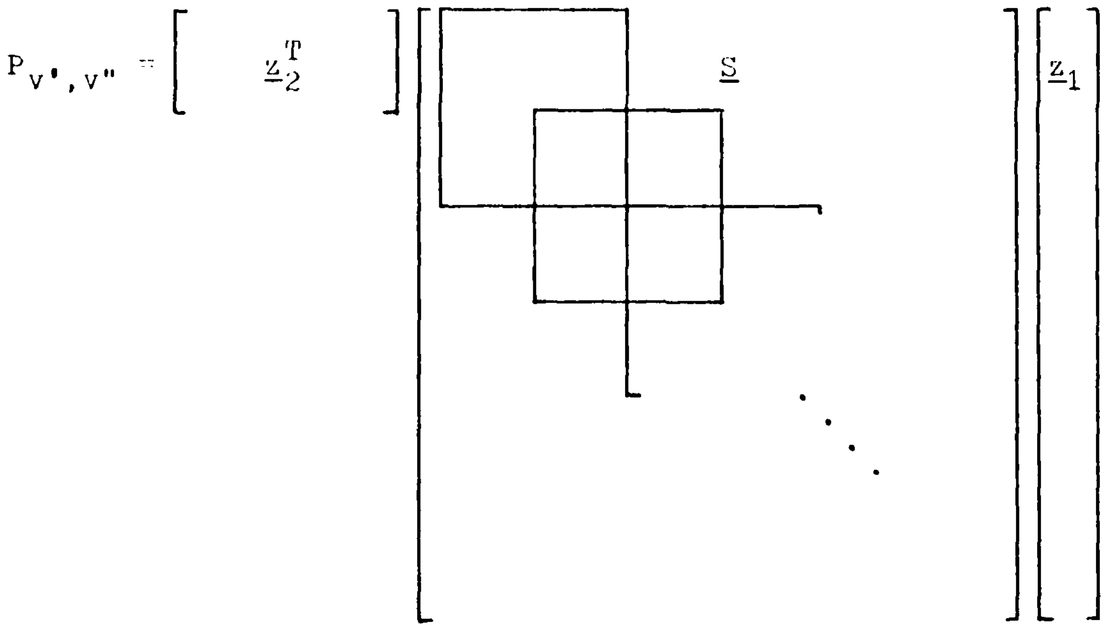
The code is written in standard Fortran 77 and should be portable. A few Harris utility routines are used, but they do not affect any calculation. Therefore, they may be replaced by similar routines found on other machines or left out of the programs. Standard routines from the IMSL (11) library are used. These routines may also be replaced if not available.

The Franck-Condon factor (Eq (75)) is:

$$P_{v',v''} = [\underline{z}_2^T \; \underline{S} \; \underline{z}_1]^2 \tag{77}$$

This method requires that the two wave functions be developed from identical grids, and therefore were derived by solving Eq (62) using the same matrix $\underset{\sim}{S}$.

$$P^{\frac{1}{2}}_{v',v''} = \underline{z}_2^T \, \underline{S} \, \underline{z}_1$$



$\underline{z}_2^T$ = a wave function of one electronic state

$\underline{z}_1$ = a wave function of the other electronic state

$\underline{S}$ = the matrix $\underline{S}$ of Fig. II-7 and Eqs (49) and (62)

Fig. II-11. The Franck-Condon Calculation in Matrix Form

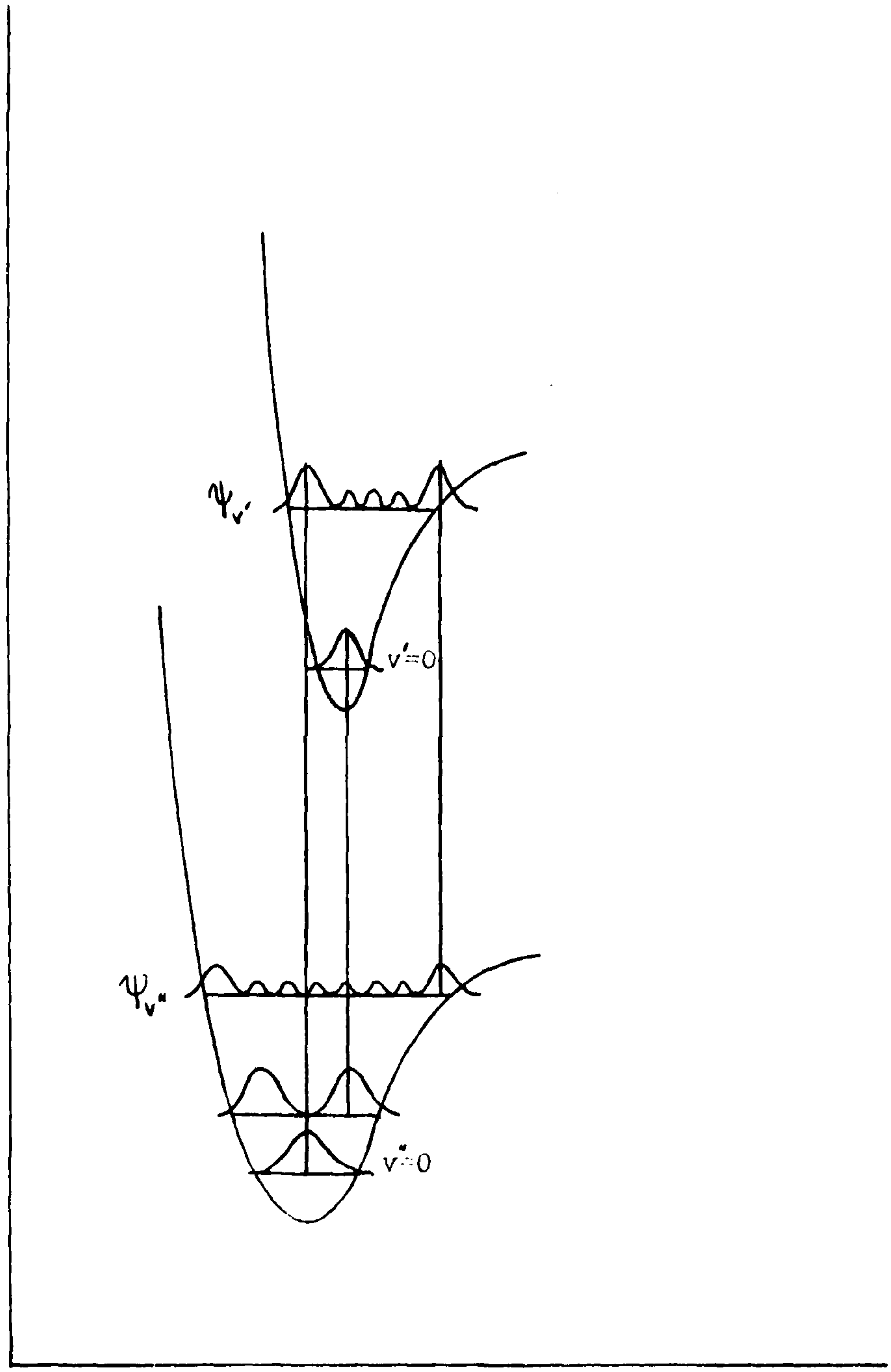


Fig. II-10. Wave Function Overlap

The development of Franck-Condon factors depends on the Born-Oppenheimer approximation (10:199) that the wave function is separable into an electronic and vibrational wave function $\psi = \psi_e \psi_{vib}$. Also, the electron is assumed to change states so quickly, compared to vibrational motion, that the nuclei have nearly the same position and velocity before and after the transition (10:199).

The inner product in Eq (75) is calculated in the same manner as the wave functions were normalized. In fact, the same matricies can be used. The difference arises since two different vectors representing the wave function are involved. The problem illustrated in Figure II-7 is changed to that in Figure II-11.

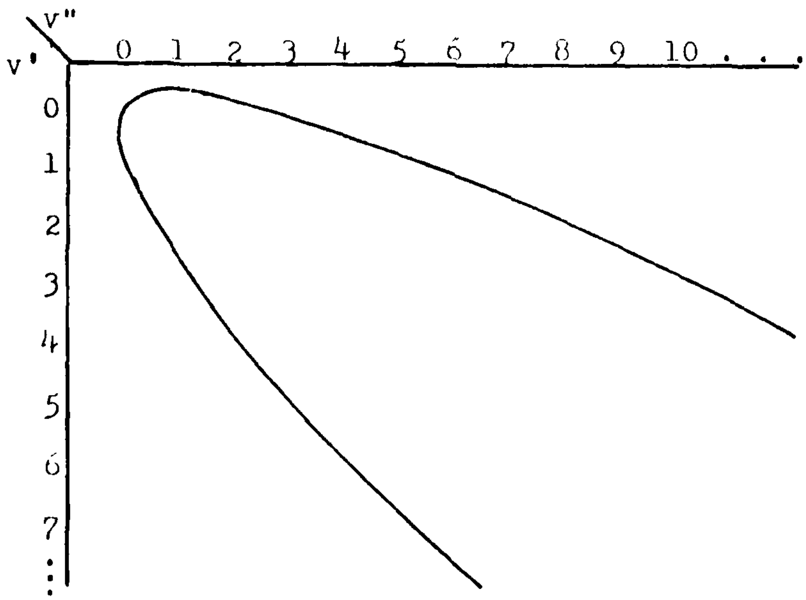


Fig. II-9. Franck-Condon Parabola

(49) illustrated in Figure II-7 as $\underline{z}^T\underline{S}\ \underline{z}$. This inner product is constrained to be one (i.e., N must be 1) by the Lagrangian multiplier constraint of Eq (63) as $\underline{z}^T\underline{S}\ \underline{z} = 1$, which is used to define the eigenvalue problem.

Calculation of Franck-Condon Factors

The wave functions for the different vibrational states of each electronic state are now used to calculate the transition probability between two states $\psi v'$ and $\psi v''$. This probability is called the Franck-Condon factor and is defined as (22:119):

$$P_{v',v''} = \left|\int \psi_{v'}^{+} \psi_{v''} dr\right|^2 \tag{75}$$

There are two notable properties of the Franck-Condon factor. First, the factors are normalized so that (22:120):

$$\sum_{v'} P_{v'',v'} = \sum_{v''} P_{v',v''} = 1 \tag{76}$$

Second, the maximum transition probabilities lie along a parabola-like curve in the array of factors (10:196). This is depicted in Figure II-9. The transition probability is highest for the wave functions which overlap the best. As Figure II-10 illustrates, there are two wave functions $\psi v''$ for which overlap is a maximum with $\psi v'$. This occurs for all but the v = 0 wave function where there is only one maximum. This corresponds to the vertex of the Condon parabola.

Once recovered from $\underline{y}$ and $\underline{L}$, the vector $\underline{z}$ contains the values and slopes of the function $U_{(r)}$ at each node r_i. The values of $U_{(r)}$ are the values of the one dimensional wave function. These wave functions should still be normalized if the vectors $\underline{y}$ were normalized, since the recovery processes were unitary. However, the wave functions are checked for normalization before using them to calculate Franck-Condon factors. The original three-dimensional (r, θ, ϕ) wave function $\psi_{(r)}$ is not needed since symmetry is assumed and $U_{(r)}$ is the one-dimensional (r) wave function.

The eigenvectors $\underline{z}$ should be orthonormal since the transformations from eigenvector $\underline{y}$ were unitary and $\underline{y}$ is orthonormal. The normality of the function $U_{(r)}$, as described by the vector $\underline{z}$, is checked by calculating a normalization constant N from the inner product of $U_{(r)}$ as:

$$|N|^2 = \left(\int_0^\infty dr U^+_{(r)} U_{(r)}\right)^{-1} \tag{72}$$

If $U_{(r)}$ is normalized, N = 1. If $U_{(r)}$ is not normalized, it can be made so $(U_{N(r)})$ by:

$$U_{N(r_i)} = NU_{(r_i)} \tag{73}$$

The inner product of $U_{(r)}$ was originally defined as

$$\langle U^+_{(r)} | U_{(r)} \rangle = \int_0^\infty dr U^2_{(r)} \tag{74}$$

in the denominator of Eq (25). In matrix form, Eq (74) is Eq

The matrix $\underline{X}$ is built similarly. Since $\underline{Z} = \underline{X}\ \underline{L}^T$, Z_{ij} is also

$$Z_{ij} = \sum_{k=j-3}^{j} X_{ik} L^T_{kj} \tag{68}$$

The sum is again limited since $\underline{L}^T$ has only four non-zero diagonals also. Remembering that $L^T{}_{kj} = L_{jk}$, expanding Eq (68) and inverting yields

$$X_{ij} = \frac{(Z_{ij} - X_{i(j-3)} L_{j(j-3)} - X_{i(j-2)} L_{j(j-2)} - X_{i(j-1)} L_{j(j-1)})}{L_{jj}} \tag{69}$$

The eigenvectors of Eq (65) are the energy levels desired, but the eigenvectors $\underline{y}$ do not contain direct information about $U_{(r)}$. This information can be recovered though.

Since $\underline{y}$ is related to $\underline{z}$ by $\underline{y} = \underline{L}^T\ \underline{z}$, the element y_{ij} is

$$\begin{aligned} y_{ij} &= \sum_{k=i}^{i+3} L^T_{ik} z_{kj} \\ &= \sum_{k=i}^{i+3} L_{ki} z_{kj} \end{aligned} \tag{70}$$

Expansion and inversion of Eq (70) yields:

$$z_{ij} = \frac{(y_{ij} - L_{(i+3)i} z_{(i+3)j} - L_{(i+2)i} z_{(i+2)j} - L_{(i+1)i} z_{(i+1)j})}{L_{ii}} \tag{71}$$

$$0 = \underline{H}\underline{z} - \lambda \underline{L}\underline{L}^T \underline{z}$$

$$0 = \underline{L}^{-1}\underline{H}\ \underline{L}^{T-1}\underline{L}^T\underline{z} - \lambda \underline{L}^{-1}\underline{L}\underline{L}^T\underline{z}$$

$$\underline{y} = \underline{L}^T\underline{z}$$

$$0 = \underline{L}^{-1}\underline{H}\underline{L}^{T-1}\underline{y} - \lambda\underline{y}$$

$$\underline{X} = \underline{L}^{-1}\underline{H}\underline{L}^{T-1} \tag{65}$$

$$0 = \underline{X}\underline{y} - \lambda\underline{y}$$

The decomposition of $\underline{S}$ is the same used to decompose $\underline{A}$ in the previous section, and will not be presented again. It should be noted though that $\underline{L}$ is a lower triangular, banded matrix.

The matrix $\underline{X}$ is derived in the following two step process. First a matrix $\underline{Z}$ is found where $\underline{Z} = \underline{L}^{-1}\ \underline{H}$. Then $\underline{X}$ is related to $\underline{Z}$ by $\underline{X} = \underline{Z}\ \underline{L}^{T-1}$.

The first step is written $\underline{H} = \underline{L}\ \underline{Z}$ where the member H_{ij} of $\underline{H}$ is:

$$H_{ij} = \sum_{k=i-3}^{i} L_{ik}Z_{kj} \tag{66}$$

The sum over k is limited since $\underline{L}$ has only four non-zero diagonals. Expanding Eq (66) and inverting yields

$$Z_{ij} = \frac{(H_{ij} - L_{i(i-3)}Z_{(i-3)j} - L_{i(i-2)}Z_{(i-2)j} - L_{i(i-1)}Z_{(i-1)j})}{L_{ii}} \tag{67}$$

$$\frac{\partial \epsilon}{\partial \underline{z}^T} = 0 \tag{64}$$

yields $\underline{H}\ \underline{z} - \lambda\ \underline{S}\ \underline{z} = 0$.

The eigenvalues of Eq (62) are the system's expected energy levels. The eigenvectors contain the value of the function $U_{(r)}$ and its slope at each node r_i.

The solution of Eq (62) requires significant computer resources for large matricies $\underline{H}$ and $\underline{S}$. AFIT does not possess computer code to take full advantage of the banded nature of the matricies involved. Over 96% of the members of $\underline{H}$ and $\underline{S}$ are zero, and only half of the 4% non-zero members need be involved in the solution due to symmetry.

A more efficient solution is now presented that allows the number of operations to be minimized for this problem. The generalized eigenvalue problem $\underline{H}\ \underline{z} - \lambda\ \underline{S}\ \underline{z} = 0$ is transformed to the standard eigenvalue problem $\underline{X}\ \underline{y} - \lambda\ \underline{y} = 0$. AFIT possesses computer code to efficiently solve this problem.

First, the Cholesky decomposition is used on $\underline{S}$ since it is symmetric, positive definite (28:229). A matrix $\underline{L}$ is found such that $\underline{S} = \underline{L}\ \underline{L}^T$. Then the problem evolves as follows:

The extrema of E occur when its first derivatives are zero, that is when:

$$0 = \frac{\partial E}{\partial z^T} \tag{59}$$

$$0 = \frac{Hz}{z^T Sz} - \frac{(z^T Hz)Sz}{(z^T Sz)^2} \tag{60}$$

$$0 = \frac{(z^T Sz)Hz - (z^T Hz)Sz}{(z^T Sz)^2} \tag{61}$$

$$0 = \underline{Hz} - \lambda \underline{Sz} \tag{62}$$

where $\lambda = (\underline{z}^T\ \underline{H}\ \underline{z})/(\underline{z}^T\ \underline{S}\ \underline{z}) = E$, the energy eigenvalues. Eq (62) is the generalized eigenvalue problem where $\underline{S}$ is a positive definite matrix. Notice that both $\underline{H}$ and $\underline{S}$ are real, band symmetric matricies.

An alternative approach arrives at the same generalized eigenvalue problem. The method of Lagrangian multipliers can be used to impose the normalization constraint ($\underline{z}^T \underline{S}\ \underline{z} - 1 = 0$). Then the problem becomes:

$$\epsilon = \mathrm{ext}\left\{\underline{z}^T \underline{Hz} - \lambda(\underline{z}^T \underline{Sz} - 1)\right\} \tag{63}$$

where λ is a set of multipliers. Then taking the first derivative equal to zero

$$I_{21} = \begin{bmatrix} U_0 & U_0' & U_1 & U_1' \end{bmatrix} \chi \begin{bmatrix} 695520 & 70560h & 82080 & -23040h \\ 70560h & 10080h^2 & 15840h & -4320h^2 \\ 82080 & 15840h & 47520 & -11520h \\ -23040h & -4320h^2 & -11520h & 2880h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_0' \\ U_1 \\ U_1' \end{bmatrix} \qquad \chi = \frac{V_0 h}{10!} \tag{54}$$

$$I_{22} = \begin{bmatrix} U_0 & U_0' & U_1 & U_1' \end{bmatrix} \chi \begin{bmatrix} 139680 & 23040h & 50400 & -12960h \\ 23040h & 4320h^2 & 11520h & -2880h^2 \\ 50400 & 11520h & 61920 & -12960h \\ -12960h & -2880h^2 & -12960h & 2880h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_0' \\ U_1 \\ U_1' \end{bmatrix} \qquad \chi = \frac{(3V_0 + V_0' h)h}{10!} \tag{55}$$

$$I_{23} = \begin{bmatrix} U_0 & U_0' & U_1 & U_1' \end{bmatrix} \chi \begin{bmatrix} 47520 & 11520h & 82080 & -15840h \\ 11520h & 2880h^2 & 23040h & -4320h^2 \\ 82080 & 23040h & 695520 & -70560h \\ -15840h & -4320h^2 & -70560h & 10080h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_0' \\ U_1 \\ U_1' \end{bmatrix} \qquad \chi = \frac{V_1 h}{10!} \tag{56}$$

$$I_{24} = \begin{bmatrix} U_0 & U_0' & U_1 & U_1' \end{bmatrix} \chi \begin{bmatrix} 61920 & 12960h & 50400 & -11520h \\ 12960h & 2880h^2 & 12960h & -2880h^2 \\ 52400 & 12960h & 139680 & -23040h \\ -11520h & -2880h^2 & -23040h & 4320h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_0' \\ U_1 \\ U_1' \end{bmatrix} \qquad \chi = \frac{(3V_1 - V_1' h)h}{10!} \tag{57}$$

The problem originally described by Eq (21) can now be written as:

$$E = \text{ext}\left\{ \frac{\underline{z}^T \underline{H}\ \underline{z}}{\underline{z}^T \underline{S}\ \underline{z}} \right\} \tag{58}$$

where $\underline{H} = \underline{T} + \underline{V}^1 + \underline{V}^2 + \underline{V}^3 + \underline{V}^4$ is the sum of matricies describing the system's kinetic and potential energy. Matrix $\underline{T}$ is built from the matricies representing I_1, and the $\underline{V}^\alpha$'s are built from the matricies representing the $I_{2\alpha}$'s in the same manner as $\underline{S}$ was built from I_3 in Figure II-7.

File XDIA

```
MS
FR ALL
AS 11=&1
AS 12=&2
AS 13=&3
AS 14=IN14
AS 15=DIARES
AS 16=DIAPOT
AS 17=DIAWAV
AS 99=%13
AS 6=%13
PR Program DIATOM is now executing . . .
DIA
PR and is now done.
ME

to use enter:  XDIA IN11 IN12 OUT13
```

Fig. III-2. Program Execution Macro

Program DUNHAM

The program DUNHAM calculates energy levels by the Dunham Eq (1) given in Chapter II. These levels may be needed to execute DIATOM.

DUNHAM accepts up to one hundred coefficients Y_{ij} where i and j range from 0 to 9. These coefficients are used to calculate up to 676 energy levels T(v,J) (v = 0 to 26; J = 0 to 26) using Eq (1). All levels may be shifted a uniform amount so the lowest level T(0,0) is any value the user desires. The user controls the number of energy levels calculated with data elements VIBLMT and ROTLMT. These data elements are the largest allowed values of v and J. The

number of energy levels written to the output file is limited by the data element LVLLMT. These levels are written in the format required by DIATOM.

DUNHAM is composed of the sub-program modules listed in Table III-1. The Harris routines (noted by a '*') are not required for energy calculations.

The program flow is presented in Appendix A. The source code is in Appendix B. DUNHAM first calls HDRDUN which uses BTIME and STIME to keep track of CPU use and how long the program runs. Then HDRDUN opens the output listing file (unit 13) and writes the listing header.

RDRDUN is then called to read all input data from unit 11. The input file (unit 11) contains fixed form, key-worded records (card images) as shown in Figure III-3. Data records are identified by a '>' in column 1. All other records are ignored as user comments. Each data record must contain a valid key word in columns 2 to 4. The valid key words and related data elements are given in Table III-2. The value of all data elements except HDR must be in columns 6 to 20. The character string used for HDR must be in columns 6-35. All real data elements must contain a decimal point. For example, '>Y10=100' is invalid, whereas '>Y10=100.0' is valid. Integer data elements may not have a decimal.

Table III-1

Program DUNHAM Modules

MODULE	TYPE	DESCRIPTION
DUNHAM	Main	The main module - calculates the energy levels.
HDRDUN	Subroutine	Opens output listing file.
BTIME *	"	Starts CPU use statistics.
STIME *	"	Starts wall clock use statistics.
DATE *	"	Returns the current date.
TIME *	"	Returns the current time.
USERNO*	"	Returns the user's ID.
RDRDUN	"	Controls all input.
TRLDON	"	Closes the output listing file.
ETIME *	"	Stops CPU use statistics.
WTIME *	"	Stops wall clock use statistics.

* Harris Routines (9)

```
**** JEFF + CONRAD'S DATA -- PBO X -- 15 OCT 84
>ROT=0
>VIB=25
>LVL=25
>HDR=PBO X state
>DEQ=0.0
>Y00=0
>Y10=722.687
>Y20=-3.613
```

Data Elements

Key Words

Fig. III-3. Program DUNHAM Input File

Table III-2

Program DUNHAM Key Words for Input Data

Key Word	Data Element Type	Data Element
VIB	Integer	Vibrational quantum number - upper limit.
ROT	Integer	Rotational quantum number - upper limit.
LVL	Integer	Total Number of energy levels to be written to the output file.
HDR	Character	A label/comment written to the output file.
DEQ	Real	The energy value of the dissociation limit.
Yij	Real	A Dunham coefficient (e.g. Y10 is the key word for Y_{10}).

After returning from RDRDUN, DUNHAM calculates each term of Eq (1) and adds it to the variable SUM. Once all values of Eq (1) are collected, the value of the energy level is transferred to the variable T. After all allowed energy levels T have been calculated, they are shifted so the dissociation energy level value is the value input on the 'DEQ' record. The shifted energy levels are then written to the output file (unit 14) in the format required for DIATOM.

Finally, TRLDUN prints the run statistics (CPU use and execution time) and closes the output listing.

Program EFIT

The program EFIT uses measured transition lines in a weighted least squares calculation of observed energy levels. These energy levels are used as input to DIATOM.

EFIT accepts 62,500 transition lines involving 250 energy levels. A maximum of ten electronic states may be entered with up to 25 levels for each state. These levels do not have to be the first 25 (v = 0 to 24), but can be any level as long as they are properly labelled on input. The user can shift all levels so that the lowest level of the lowest state has a specified value. Each level calculated is written to an output file in the format required by DIATOM.

EFIT is composed of the sub-program modules presented in Table III-3. The program flow is depicted in Appendix C and the source code is in Appendix D.

Table III-3

Program EFIT Modules

MODULE	TYPE	DESCRIPTION
EFIT	Main	The main module - does the least squares fit.
HDREFT	Subroutine	Opens the output listing file.
BTIME *	"	Starts CPU use statistics.
STIME *	"	Starts wall clock use statistics.
DATE *	"	Returns the current date.
TIME *	"	Returns the current time.
USERNO*	"	Returns the user's ID.
RDREFT	"	Controls all input.
LVLEFT	"	Calculates the absolute energy level number for an energy level relative to an electronic state.
TRLEFT	"	Closes the output listing file.
ETIME *	"	Stops CPU use statistics.
WTIME *	"	Stops wall clock use statistics

* Harris Routines (9)

EFIT first calls HDREFT which performs the same functions described for HDRDUN in program DUNHAM.

Then RDREFT opens and reads all records in the input file (unit 11). The input technique is the same described for RDRDUN of DUNHAM. The key words are six characters long and are in colums 2-7 of each record. Table III-4 gives the acceptable key words and related data elements.

Figure III-4 is an example of a valid input file. In this example, the first state (STAT01) is the X or ground electronic state, the second (STAT02) the a or first electronically excited state, and so on. Notice that data for the X state vibrational levels v = 0 to 8 involved in the following records are indicated by the record keyworded LVLS01. Also notice that the b state has only one level involved (v = 4). Missing levels are allowed. The program does not require every level to be represented by the data in the range v = (lowest value) to v = (highest value) for each state. The record keyworded SHIFTS contains the value of the lowest level of the lowest state on output. All other levels will be shifted so this occurs. Transition data is given by the remaining records key-worded ABBCDD. These records contain the transition line observed (cm^{-1}) and optionally the least squares weighting factor ($0 \leq w \leq 1.0$) separated by a ';' for the transition from vibrational level BB of electronic state A to level DD of state C. If the weighting

Table III-4

Program EFIT Key Words for Input Data

KEY WORD	DATA ELEMENT TYPE	DATA ELEMENT
STATxx	Character	A single character symbol for the electronic state xx; e.g., STAT01=x assigns the symbol "x" to the first state.
LVLSxx	Integer	A series of integer numbers (1 or 2 digit) indicating which energy levels V=? are involved in the least squares fit for electronic state xx symbolized by STATxx.
SHIFTS	Real	The value of the lowest level of the lowest state.
ABBCDD 1;w	Real	Two data elements - the value (1) of the transition observed between vibrational level BB of electronic state A and level DD of state C and its weighting factor (W=0 to 1.0). The two values are separated by a ";".

```
********************************************************************
***
***          This is an input file for EFIT
***
********************************************************************
***
*** This file contains Ritchey's data for Pb + O2(3sigma)   pg. 45
***
>STATO1=X
>STATO2=a
>STATO3=b
>STATO4=A
>STATO5=B
>STATO6=C
>LVLSO1=0 1 2 3 4 5 6 7 8
>LVLSO2=0 1 2 3 4 5 6 7 8 9
>LVLSO3=4
>LVLSO4=0 1 2 3 4 6 8
>LVLSO5=0 1 2 3 4 5
>LVLSO6=0
>SHIFTS=-30854.15
>a08X00=19584.4
>a07X00=19154.2
>a06X00=18702.4
>a05X00=18257.2
>a09X03=17883.0
>a04X00=17801.5
>a05X01=17539.2
>a03X00=17334.6
>a04X01=17084.1
>b04X08=12416.5
>A08X01=22483.5
>A06X01=21675.0
>A04X00=21519.3
>A03X00=21047.8
>A06X02=20942.0
```

Electronic state a contributes 10 levels, v=0 to 9, to the least squares fit.

No data will be entered for state b levels v=0 to 3 or v greater than 4.

The transition line observed between state a v=4 and state X v=0 is 17801.5 inverse cm. Weighting factor defaults to 1.

Fig. III-4. Program EFIT Input File

factor is missing on input it defaults to 1.0, the highest weighting allowed. The order of the keywords is important and should match that given by Figure III-4. For example, the LVLS01 record can not occur before STAT01 record.

Control now passes back to the main module. EFIT builds the matrix $\underline{A}$ (Figure II-3) of the least squares equation $\underline{A}\ \underline{x} = \underline{b}$. $\underline{A}$ is stored in full matrix storage mode such that A(I,J) corresponds to A_{ij}. The matrix $\underline{b}$ is built next. Then the problem is changed to $\underline{Ly} = \underline{b}$ where $\underline{A} = \underline{L}\ \underline{L}^T$ and $\underline{y} = \underline{L}^T\ \underline{x}$ by the Cholesky decomposition Eq (10). The solution, $\underline{y}$, is found through Eq (12) and transformed to $\underline{x}$ by Eq (15) after a consistency check is made (Eq (13)). The matrix $\underline{x}$ contains the unshifted energy levels. Then all levels are shifted so the lowest level is at 0. The levels are then written to the output listing. Finally, they are shifted again so that the lowest level is at the value input by the SHIFTS records. These final energy level values are written to an output file suitable for DIATOM.

A possible source of confusion is the energy level labelling used by EFIT. EFIT gives each level an absolute level number, while also maintaining a level number relative to the electronic state for the user's convenience. This is illustrated in Table III-5. EFIT uses absolute level numbers in array references and calculations.

Table III-5

Relative and Absolute Energy Level Numbering

STATE NUMBER	STATE SYMBOL	RELATIVE NUMBER	ABSOLUTE NUMBER
01	X	1	1
		2	2
		3	3
02	a	1	4
		2	5
03	b	4	6
04	A	0	7
		1	8
		2	9
		3	10
		4	11
		8	12

Program DIATOM

The program DIATOM finds a set of parameters for the potential energy model that best fits the observed energy levels in a weighted least squares sense. Then DIATOM finds the normalized wave functions for the calculated energy levels. These wave functions are needed for the program FCFACT.

The execution of DIATOM may be modified by keyworded records 'RM' and 'RW'. If RM = 1, DIATOM finds the potential energy parameters that best fit the energy levels. If RW = 1, DIATOM calculates the wave functions.

DIATOM accepts up to 25 observed vibrational energy levels for a specific electronic state. However, these levels must be consecutive and start at v = 0. Missing

levels may be marked by assigning it a value of zero. Then no comparison is made by DIATOM to the calculated value. The grid used by DIATOM may have up to 100 steps (101 nodes). The potential energy model may use up to ten constants and ten parameters. Each parameter is associated with an upper and lower limit.

DIATOM is composed of the modules listed in Table III-6. The program flow is presented in Appendix E and the source code is in Appendix F.

DIATOM first calls HEADER which performs the same functions HDRDUN of program DUNHAM performed.

Then READER opens and reads all records from the input files (unit 11, 12, and 14). The input technique described for RDRDUN of program DUNHAM applies. Unit 11 contains data that control the execution of DIATOM. Valid key words and data elements are listed in Table III-7. The keywords must be in columns 2-3. An example of a valid unit 11 input file is given in Figure III-5. Records key worded 'Cx' contain constants used by the potential energy model. Records key worded 'Lx', 'Px', and 'Ux' contain, respectively, values of the lower limit, initial value, and upper limit of a potential energy parameter.

Unit 12 contains data related to the observed energy levels. Valid key words and data elements are given in Table III-8, and an example of a valid unit 12 input file is

Table III-6

Program DIATOM Modules

MODULE		TYPE	DESCRIPTION
DIATOM		Main	The main program module.
HEADER		Subroutine	Opens the output listing file.
BTIME	*	"	Starts CPU use statistics.
STIME	*	"	Starts wall clock use statistics.
DATE	*	"	Returns the current date.
TIME	*	"	Returns the current time.
USERNO	*	"	Returns the user's ID.
READER		"	Controls all input.
MINUM		"	Finds the minimum of FUN.
FUN		Function	The sum of weighted residuals squared.
POTENT		Subroutine	Returns potential energy value and slope.
EIGEN		"	Solves $\underline{H}v - \underline{S}v = 0$.
GETL		"	Cholesky decomposition $\underline{S} = \underline{LL}^T$.
FOLDZ		"	Finds $\underline{Z}$ where $\underline{H} = \underline{LZ}$.
FOLDX		"	Finds $\underline{X}$ where $\underline{X} = \underline{ZL}^{T-1}$.
EHOUSS	#	"	Changes $\underline{X}$ to tridiagonal form.
EQRT1S	#	"	Finds eigenvalues of tridiagonal $\underline{X}$.
EQRT2S	#	"	Finds eigenvalues and eigenvectors of tridiagonal $\underline{X}$.
EHOBKS	#	"	Finds eigenvectors of $\underline{X}$ from those of tridiagonal $\underline{X}$.
UNFOLD		"	Finds eigenvectors $\underline{v}$ from those of $\underline{X}$.
PUTRND		"	Stores random number generator values.
WAVE		"	Finds the wave functions.
ICSCCU	#	"	Fits a cubic spline to a function.
NORMAL		"	Normalizes the wave functions.
VMULQF	#	"	Matrix multiplication.
VMULFF	#	"	Matrix multiplication.
PUTWAV		"	Stores wave functions in a file.
OUTPUT		"	Writes output listing and data.
GETPUT		"	Calculates 1000 potential energy values.
PRNTER		"	Creates the listing.
PLTRES		"	Creates residual plot file.
PLTPOT		"	Creates potential plot file.
TRAILR		"	Closes the output listing file.
ETIME	*	"	Stops CPU use statistics.
WTIME	*	"	Stops wall clock use statistics.

* Harris Routines (9)
\# IMSL Routines (11)

Table III-7

Program DIATOM Key Words for Input Data - Unit 11

KEY WORD	TYPE	DATA ELEMENT
BG	Real	The leftmost (beginning) mode of the grid.
EN	"	The right most (ending) mode of the grid.
NE	Integer	The number of grid elements (steps).
IS	"	The number of steps MINUM may take.
IP	"	The modulus of the steps at which MINUM printing is required. If negative no information is printed.
JR	"	Random step frequency (MINUM).
JG	"	Gradient step frequency (MINUM).
JA	"	Average step frequency (MINUM).
JJ	"	Jump step frequency (MINUM).
FR	"	A flag for creating residual plot file (FR=1 yes; FR=0 no).
FP	"	A flag for creating potential.
RM	Integer	A flag controlling execution of MINUM. If RM=1, MINUM is used. If RM=0, MINUM is not used.
RW	"	A flag controlling execution of WAVE. If RW=1, WAVE is used. If RW=0, WAVE is not used.
HB	Real	The value of Planck's Constant h.
MU	"	The molecule's reduced mass.
Cx	"	"x" = 0 to 9. The value of one of 10 constants available to the potential energy model.
Px	"	"x" = 0 to 9. The initial value of one of ten paramenters available to the potential model. MINUM changes these.
Lx	"	Lower limit of Px.
Ux	"	Upper limit of Px.

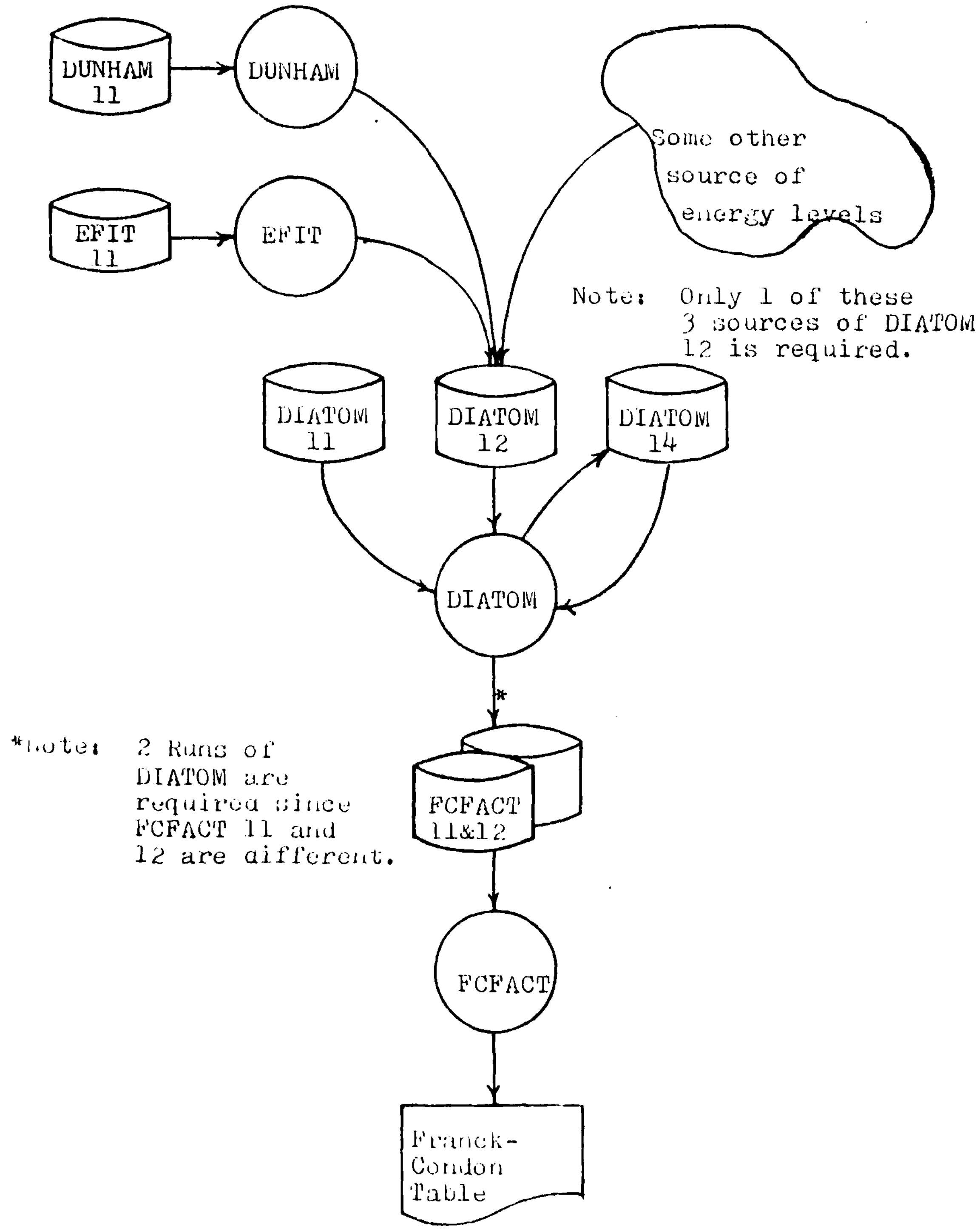


FIG. III-11. Overview of Program Relationships

CLCFCF builds the matricies of Figure II-11 and multiplies them in a manner similar to that used by NORMAL of DIATOM. The matrix product is the inner product between the normalized wave functions. This product is squared and stored in the two dimensional array FACTOR as the Franck-Condon factor.

OUTFCF prints the Franck-Condon table in two sections after the first loop is done. Then, TRLFCF stops the run statistics and closes the output listing file (unit 13).

Program Relationships

The programs DUNHAM, EFIT, DIATOM and FCFACT are related as shown in Figure III-11. The user may use DUNHAM, EFIT, or some other source of energy level data to build the input file (unit 12) for DIATOM. DIATOM in turn builds one of two input files for FCFACT. The second FCFACT input file is built by running DIATOM for a different electronic state (a new set of energy levels and potential parameters).

```
>LABL=Single Harmonic Oscillator
>S  1    0.000000        0.000000         0.000000        0.1273469
>S  2    0.000000        0.000000         0.6157434E-02   0.3838401E-03
>S  3    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S  4  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S  5    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S  6  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S  7    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S  8  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S  9    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 10  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 11    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 12  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 13    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 14  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 15    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 16  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 17    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 18  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 19    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 20  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 21    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 22  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 23    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 24  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 25    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 26  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 27    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 28  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
 .
 .
 .
```

Matrix S Data

```
>S 69    0.000000        0.4408163E-01    0.3638484E-02   0.2546939
>S 70  -0.3638484E-02  -0.2878800E-03     0.000000        0.7676801E-03
>S 71    0.000000        0.4408163E-01    0.3638484E-02   0.1273469
>S 72  -0.3638484E-02  -0.2878800E-03   -0.6157434E-02    0.3838401E-03
>   0    1 -0.1293752E-10
>   0    2 -0.2054280E-09
>   0    3 -0.8461519E-10
>   0    4 -0.1223209E-08
>   0    5 -0.1591142E-09
>   0    6 -0.3602358E-08
>   0    7 -0.4675904E-09
 .
 .
 .
```

Eigenvector Data

Fig. III-10. Program FCFACT Input File

Valid key words (columns 2-5) and data elements are listed in Table III-11. An example of a valid input file for units 11 and 12 is given in Figure III-10. The 'LABL' record contains a 72 character label of the wave function set. All records with a "S" in column 2 contain data belonging to the matrix $\underline{S}$ of $\underline{z}_2^T \ \underline{S} \ \underline{z}_1$. The "SXXX" record contains the "XXX" row of matrix $\underline{S}$ in band symmetric form. The key word of the remaining "XXXX" records is the energy level number. Columns 6-9 contain a sequential record number relative to the energy level. Columns 10-24 contain the value of the eigenvector corresponding to the number in columns 6-9.

Next, FCFACT sets LUNIT = 12 and calls RDRFCF to read all data concerning the upper (v') state.

The Franck-Condon factors are calculated by CLCFCF which is called for each combination of wave functions.

Table III-11

Program FCFACT Key Words for Input Data

KEY WORD	TYPE	DATA ELEMENT
LABL	Character	A 72 character label of the wave function set.
"Sxxx"	Real	4 values belonging to the "xxx" row of matrix $\underline{S}$ in band symmetric form. (e.g. "S011" contains S(11,1), S(11,2), S(11,3), and S(11,4)).
"XXXX"	Real	"XXXX" = "1" to "25". The data for energy level "XXXX".

Table III-10

Program FCFACT Modules

MODULE		TYPE	
FCFACT		Main	The main module.
HDRFCF		Subroutine	Opens the output listing file.
BTIME	*	"	Starts CPU use statistics.
STIME	*	"	Starts wall time use statistics.
DATE	*	"	Returns current date.
USERNO	*	"	Returns the user's ID.
RDRFCF		"	Reads one set of wave functions.
CLCFCF		"	Calculates the FCF for two wave functions.
OUTFCF		"	Prints the FCF table.
TRLFCF		"	Closes the output listing file.
ETIME	*	"	Stops CPU use statistics.
WTIME	*	"	Stops wall time use statistics.

* Harris Routines (9)

Finally, TRAILR stops the run statistics and closes the output file.

Program FCFACT

Program FCFACT computes the inner product given by Eq (75), (Franck-Condon factors) for two sets of wave functions. Each wave function set is the output of one run of DIATOM.

FCFACT expects both input files (units 11 and 12) to have identical matricies $\underset{\sim}{S}$ and 25 (v = 0 to v = 24) sets of wave function data. Both sets must be related to an identical grid in DIATOM. FCFACT builds a 25 by 25 table of Franck-Condon factors.

FCFACT is composed of the sub-program modules listed in Table III-10. The program flow is presented in Appendix G, and the source code is in Appendix H.

FCFACT first calls HDRFCF which performs as HDRDUN of DUNHAM did.

Then FCFACT opens units 11 and 12, sets the variable LUNIT = 11, and calls RDRFCF. This causes RDRFCF to read all data concerning the lower (v") state.

where wi is the weighting factor ($0 \leq w \leq 1$) associated with E_i.

Control is passed to PUTRND after MINUM finishes. PUTRND stores the current values of the random number generator seeds for next time DIATOM is executed.

Next, WAVE finds the normalized wave functions (eigenvectors) for each calculated energy level (eigenvalue). First, the main module sets JOBN = 1 before WAVE is called. Then WAVE calls FUN using the "best" parameters found by MINUM. This causes the eigenvectors of the standard eigenvalue problem (Eq (65)) to be computed. WAVE also writes the matrix $\underline{S}$ to unit 17 for the program FCFACT.

Then NORMAL uses IMSL routines VMULQF and VMULFF are used to multiply the matricies of Eq (77), and the normalization factor N (Eq (81)) is computed. Then the normalized value of the wave function is computed. PUTWAV then stores the eigenvectors (wave functions) in unit 17 using the format required by program FCFACT.

Then, OUTPUT calls GETPOT to calculate the value of the potential energy model at 1000 points along the grid. These values can be used for plotting the potential. PRNTER writes the potential values, parameters, and energy data to the output listing. OUTPUT also uses PLTRES and PLTPOT if DIATOM was directed to create plot files of the residual (unit 15) and potential (unit 16) data (FR = 1; FP = 1).

Then the generalized eigenvalue problem Eq (62) is solved when FUN calls EIGEN. EIGEN first calls GETL to decompose the matrix S by Eq (10). Then EIGEN uses FOLDZ and FOLDX to implement Eq's (67) and (69). This is how the generalized eigenvalue problem is transformed into the standard eigenvalue problem Eq (65). The symmetric matrix $\underline{X}$ of the standard eigenvalue problem is changed to a symmetric tridiagonal matrix $\underline{T}$ by IMSL routine EHOUSS (11) using Householder's transformation (13). Then the lowest 25 eigenvectors of $\underline{T}$ are found by IMSL routine EQRT1S (11) using a QR transformation with a Newton shift (18). If JOBN = 1, the eigenvectors of $\underline{T}$ are computed by the QR method (1) using IMSL routine EQRT2S (11). The eigenvectors of $\underline{X}$ are found from those of $\underline{T}$ by IMSL routine EHOBKS (11), (13).

Next, UNFOLD recovers the eigenvectors of the generalized eigenvalue problem Eq (62) from those found by FUN.

FUN then compares the eigenvalues $_V$ and the observed energy levels E_V by computing a residual (variable RESID):

$$\mathrm{RESID(I)} = \begin{cases} (\lambda_I - E_I) & \text{for } E_I \neq 0 \\ 0 & \text{for } E_I = 0 \end{cases} \tag{78}$$

Then the weighted sum the residuals is computed as FUN:

$$\mathrm{FUN} = \tfrac{1}{2}\sum_{i=1}^{25} (w_i^2 \times \mathrm{RESID(i)}^2) \tag{79}$$

$$
\text{Matrix } \underline{H} \text{ of order } 3 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} h_{11} \\ h_{21} \\ h_{22} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}
$$

Full Storage Mode

Symmetric Storage Mode

Fig. III-8. Symmetric Matrix Storage Mode

$$
\text{Matrix } \underline{S} \text{ of order } 5 = \begin{bmatrix} s_{11} & s_{12} & 0 & 0 & 0 \\ s_{21} & s_{22} & s_{23} & 0 & 0 \\ 0 & s_{32} & s_{33} & s_{34} & 0 \\ 0 & 0 & s_{43} & s_{44} & s_{45} \\ 0 & 0 & 0 & s_{54} & s_{55} \end{bmatrix} = \begin{bmatrix} 0 & s_{11} \\ s_{21} & s_{22} \\ s_{32} & s_{33} \\ s_{43} & s_{44} \\ s_{54} & s_{55} \end{bmatrix}
$$

Full Storage Mode

Band Symmetric Storage Mode

Fig. III-9. Band Symmetric Matrix Storage Mode

The variable JOBN controls calculation of eigenvalues (JOBN = 0) or eigenvectors (JOBN = 1). DIATOM initially sets JOBN = 0 since the eigenvectors are not needed until the best parameter set has been found by MINUM. MINUM is a routine written by Pearson and Williams (15) that computes the minimum of a real function FUN. FUN is a function of ten potential energy parameters which must be scaled so each range from 0 to 1.0. MINUM may be replaced by any non-linear minimization routine. MINUM takes random and statistically derived "best guess" steps and jumps through FUN's parameter space.

The function FUN first builds the matricies $\underline{H}$ and $\underline{S}$ of Eq (58). FUN calls POTENT once for each grid element in this process. The matrix $\underline{H}$ is stored in symmetric mode as a one dimensional array of length $n(n + 1)/2$ where n is the order of the matrix. The matrix element H_{ij} is the k^{th} element of the array where $k = (i(i - 1)/2) + j$. Due to symmetry, only the lower triangle of $\underline{H}$ $(i \geq j)$ is stored. This is illustrated in Figure III-8. Matrix $\underline{S}$ is stored in band symmetric mode, illustrated in Figure III-9, in a two-dimensional array. Only the elements on the main and sub-diagonals are stored. The matrix element S_{ij} ($i = 1$ to n; $j = i, (i - 1), (i - 2), (i - 3)$) is stored in the k, l^{th} element of the array where $k = i$; $l = 4 - (i - j)$; and $j = (k + l) - 4$.

Table III-9

Program DIATOM Key Word
Input Data - Unit 14

KEY WORD	TYPE	DATA ELEMENT
IU	Integer *6	Starting random number seed.
IX	"	Fixed random number multiplier (IX=131075).

```
************************************************************************
***        Seeds for a random number generator in MINUM
***                       called by program DIATOM
************************************************************************
>IU= 51313486833257
>IX=          131075
```

Fig. III-7. Program DIATOM Input File - Unit 14

```
* Single Harmonic Oscillator
* 24 Aug 84
>LBL=Single Harmonic Oscillator
>V00=1.0
>V01=3.0
>V02=5.0
>V03=0.0
>V04=9.0
>V05=11.0 ; .9
>V06=13.0 ; .8
>V07=15.0 ; .7
>V08=17.0 ; .6
>V09=19.0 ; .5
>V10=21.0 ; .4
```

The v=3 level is not observed and must be entered as 0.

The v=8 level is weighted 6 tenths of the other levels (v=0 to 4) which default to 1.0.

Fig. III-6. Program DIATOM Input File - Unit 12

are entered starting a v = 0 to 4 and v = 6 to 13, then 14 records 'Vxx' must be entered 'xx' = '00' to '13'. Record 'V05' must be zero since no level was observed for v = 5. This level will not be used in the energy level fits and serves only as a place holder.

Unit 14 contains values used as seed numbers for MINUM's random number generator. DIATOM replaces the seed values with new values every run. Therefore, the user need only supply this data when the program is first put on the computer. At this time both may be set to 131075. Valid key words must be in columns 2-3 and are listed in Table III-9. An example of an input file is given in Figure III-7.

Table III-8

Program DIATOM Key Words for Input Data - Unit 12

KEY WORD	TYPE	DATA ELEMENT
LBL	Character	A 72 character label used for the wave function output file.
Vxx e;w	Real	"xx" = "01" to "10". Two data elements. First is the value of the observed energy (e) level xx. Second is a weighting factor (w = 0 to 1.0) applied to that level separated by a ";".

presented in Figure III-6. Keywords must be in columns 2-4. The character string contained in the 'LBL' record is used to label the wave function output file (unit 17). Each 'Vxx' records contains the observed energy value and, optionally, a weighting factor ($0 \leq w \leq 1.0$) for the energy level v = 'xx'. The units of the energy level must be the same used for the values of Planck's constant $\hbar$ (record 'HB' in unit 11) and the system's reduced mass μ (record 'MU' in unit 11). The energy level value and the weighting factor are separated by a ';'. The weighting factor defaults to 1.0 if it is not entered. The 'Vxx' records must be consecutive in 'xx'. If 13 levels

```
* Single Harmonic Oscillator
* 24 Aug 84
>BG=-6.0
>EN=6.0
>NE=90
>IS=1
>IP=1
>JR=5
>JG=1
>JA=5
>JJ=1
>FR=0
>FP=0
>RM=0
>RW=1
>HB=1.0
>MU=1.0
>L1=1.99
>P1=2.0
>U1=2.01
>L2=-0.02
>P2=0.0
>U2=0.02
```

MINUM is not allowed to run.

WAVE is allowed to run.

Potential Energy parameter 1 starts at 2.0 and may vary from 1.99 to 2.01

Fig. III-5. Program DIATOM Input File - Unit 11

IV. Results and Discussion

This section consists of three parts. The first two parts present the validation of the programs DIATOM and FCFACT. Then a CPU use benchmark is presented of the program DIATOM in the third part. The programs DUNHAM and EFIT were tested against several sets of hand calculations. These programs were found to be accurate to within the number of significant digits supported by single precision computer operations. These results are not presented.

Validation of the Program DIATOM

The program DIATOM was tested against the analytic solutions of the single harmonic oscillator. The harmonic oscillator was chosen since the lower vibrational states of most diatomic molecules are nearly harmonic. The analytic energy levels and wave functions of the harmonic oscillator are presented in Appendix J.

DIATOM was tested using atomic units where $\hbar = \mu = 1$. The potential energy model chosen was:

$$V_{(r)} = p_1 r^2 + p_2 \tag{80}$$

The parameter p_1 is related to the scaling constant α of Appendix J by:

$$p_1 = \frac{(h\alpha)^2}{2\mu} \tag{81}$$

and to ω and μ by:

$$p_1 = \tfrac{1}{2}\mu\omega^2 \tag{82}$$

The parameter p_2 merely shifts the minimum of the potential curve. This parameter was used to make the problem of finding the correct parameter set (p_1,p_2) more difficult for MINUM.

The solution of the wave equation by DIATOM shall be referred to as WAVE. This means that the input records RM=0 and RM=1 were used in unit 11. The use of DIATOM to search through the parameter space to find the correct parameter set is referred to as MINUM. This means that unit 11 contains the records RM=1 and RM=0. WAVE was tested first.

The WAVE version of DIATOM was executed with $p_1 = 2$ and $p_2 = 0$. This means that α is one and ω is two. Then a variety of grid resolutions were used. Each grid started at $r = -6$ and ended at $r = 6$. The number of elements in the grid were varied from five to ninety. The results of these tests are presented in Tables IV-1, IV-2, and IV-3.

The energy eigenvalues for the analytic and numeric (calculated by DIATOM) cases of 25, 50, 75, and 90 element grids are presented in Table IV-1. Notice that the numerical eigenvalues approach the analytic eigenvalues as the grid becomes finer. The solutions of DIATOM converge to the analytic solution as the grid element size becomes smaller.

Table IV-1

Analytic vs. Numeric Eigenvalues
of the Single Harmonic Oscillator

v	ANALYTIC	NUMBER OF GRID ELEMENTS 25	50	75	90
0	1.0	1.000013	1.000000	1.000000	1.000000
1	3.0	3.000106	3.000002	3.000000	3.000000
2	5.0	5.000431	5.000011	5.000001	5.000000
3	7.0	7.001212	7.000032	7.000003	7.000001
4	9.0	9.002718	9.000077	9.000008	9.000003
5	11.0	11.00511	11.00016	11.00002	11.00001
6	13.0	13.00931	13.00028	13.00003	13.00001
7	15.0	15.01265	15.00047	15.00005	15.00002
8	17.0	17.02656	17.00073	17.00008	17.00003
9	19.0	19.01727	19.00108	19.00012	19.00004
10	21.0	21.07060	21.00153	21.00018	21.00007
11	23.0	23.01505	23.00211	23.00025	23.00009
12	25.0	25.12566	25.00282	25.00034	25.00013
13	27.0	27.07469	27.00368	27.00045	27.00017
14	29.0	29.11559	29.00471	29.00059	29.00022
15	31.0	31.19916	31.00593	31.00075	31.00028
16	33.0	33.18257	33.00734	33.00094	33.00035
17	35.0	35.22109	35.00898	35.00116	35.00044
18	37.0	37.32730	37.01085	37.00112	37.00054
19	39.0	39.37048	39.01297	39.00171	39.00065
20	41.0	41.39660	41.01537	41.00205	41.00078
21	43.0	43.50121	43.01805	43.00243	43.00093
22	45.0	45.62668	45.02103	45.00285	45.00110
23	47.0	47.70044	47.02432	47.00332	47.00127
24	49.0	49.77736	49.02793	49.00380	49.00144

Table IV-2

An Overview of Eigenvalue Accuracy

NUMBER OF GRID STEPS	CPU SEC.	NUMBER OF CALCULATED EIGENVALUES WITHIN X% OF ANALYTIC VALUE			
		0.005%	0.01%	0.05%	0.1%
5	1.9	0	0	0	0
10	3.3	0	0	1	1
15	5.2	0	0	3	3
20	9.0	1	2	3	5
25	14.2	2	3	6	10
30	17.0	3	5	9	12
35	25.0	4	7	12	16
40	35.6	6	9	15	20
45	51.3	8	11	20	25
50	76.2	9	14	24	25
55	121.1	11	17	25	25
60	186.4	14	20	25	25
65	273.1	15	24	25	25
70	396.3	20	25	25	25
75	575.1	21	25	25	25
80	786.2	24	25	25	25
90	1377.8	25	25	25	25

Table IV-3

An Overview of Wave Function Accuracy

NUMBER OF GRID STEPS	% AGREEMENT BETWEEN WAVE FUNCTIONS AT r=0 v=0	v=8	v=16	v=24
10	0.70	*	*	*
20	0.14	*	*	*
30	0.03	4.40	*	*
40	0.01	1.18	5.44	*
50	0.005	0.47	1.91	4.82
60	0.002	0.22	0.86	2.05
70	0.001	0.12	0.45	1.03
80	0.001	0.07	0.27	0.59
90	0.005	0.05	0.17	0.37

* The corresponding eigenvalue did not meet the 0.1% accuracy criteria.

This is expected since the problem was constructed for convergence (17:114-115).

Table IV-2 is an overview of the accuracy obtained for grids of different resolutions. DIATOM computed the 25 lowest energy eigenvalues to better than or equal to 0.005% for a grid of 90 elements (r = -6 to 6). On the other hand, all 25 eigenvalues were computed to within 0.1% accuracy for a lower resolution grid of 45 elements (r = -6 to 6). The higher resolution grid is over 26 times more expensive (1377.8 vs. 51.3 CPU seconds) to use than the lower resolution grid. This is the trade off the user must come to terms with. The computation cost must be considered when specifying the desired accuracy.

Table IV-3 presents an indication of the accuracy of the wave functions calculated by WAVE (DIATOM). The approximate wave functions vary the most from the analytic wave functions at the center of peaks or troughs. Therefore, the wave functions are compared at r = 0, the middle peak or trough of the even (v = 0,2,4,...) wave functions. This comparison is only an indication, not a true measure of the wave function's accuracy. No value was given in Table IV-3 if the accuracy of the eigenvalue was greater than 0.1%, or if the wave function did not have the correct form. Again, notice that the approximate solutions calculated by DIATOM converge to the analytic solutions as a finer, more expensive grid is used. Also notice that the more accurate wave functions are

the lower state wave functions. In all cases the v = 24 wave function was the least accurate. The best accuracy obtained for v = 24 at r = 0 was 0.37% while 0.005% accuracy was obtained for v = 0. Better accuracy should be possible with a grid of more than 90 elements.

The v = 0, 3, 7 and 9 wave functions are plotted in Figure IV-1. The dotted line is the analytic wave function and the solid line is the numeric wave function for a 65 element grid. Each numeric wave function varied no more than 0.3% (in the sense previously discussed) in absolute value from the analytic wave function. The v = 0 and 7 wave functions differ from the analytic by a minus sign. This is an artifact of the QR method used by the IMSL eigenvalue routine EQRT2S. Each eigenvector is unique to within a minus sign (11). The inversion of a wave function is of no consequence though, since the Franck-Condon factor depends on the square of the inner product.

Next, DIATOM was tested using MINUM (RM=1, RW=0) to see if the correct potential energy parameter set could be found. DIATOM was run four times in this manner using the parameters and search limits of Table IV-4. After the fourth run, MINUM chose the parameters p_1 = 2.0004 and p_2 = -0.002. These values are close to the correct values of p_1 = 2.0 and p_2 = 0. A closer fit is possible by running MINUM (DIATOM) again with a narrower search region and more grid elements.

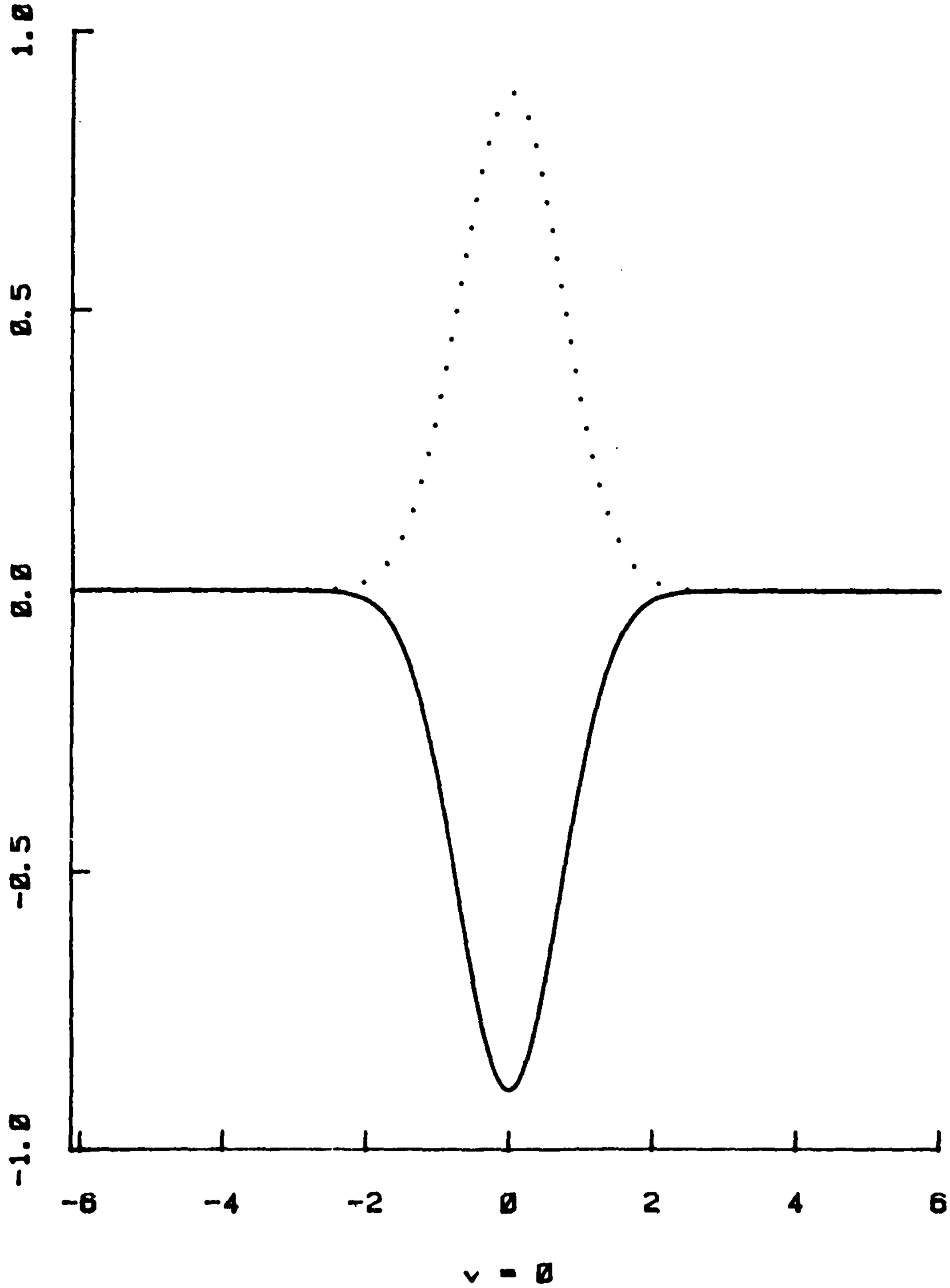


Fig. IV-1a. Selected Analytic and Numeric Wave Functions of the Single Harmonic Oscillator

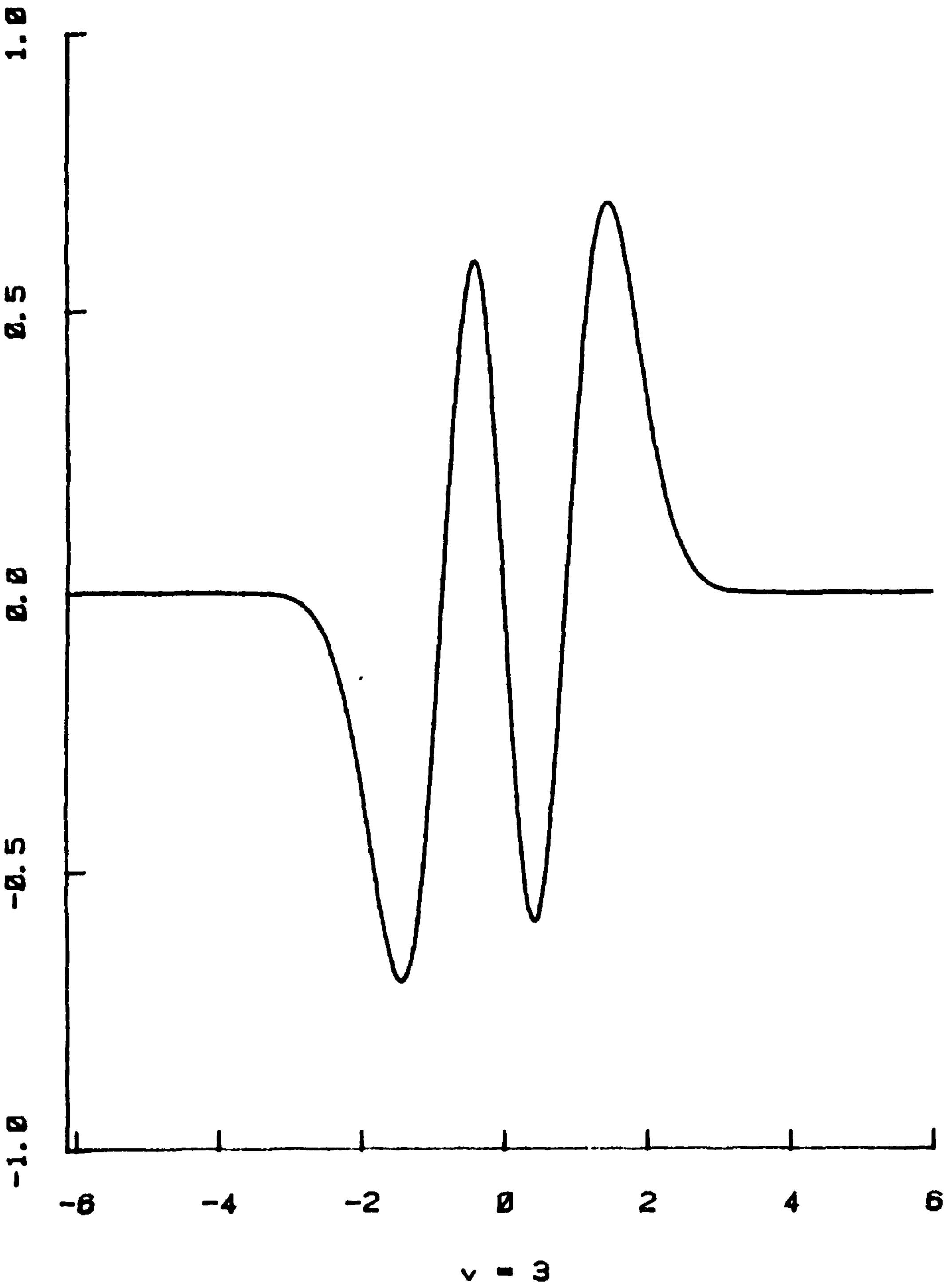


Fig. IV-1b. Selected Analytic and Numeric Wave Functions of the Single Harmonic Oscillator

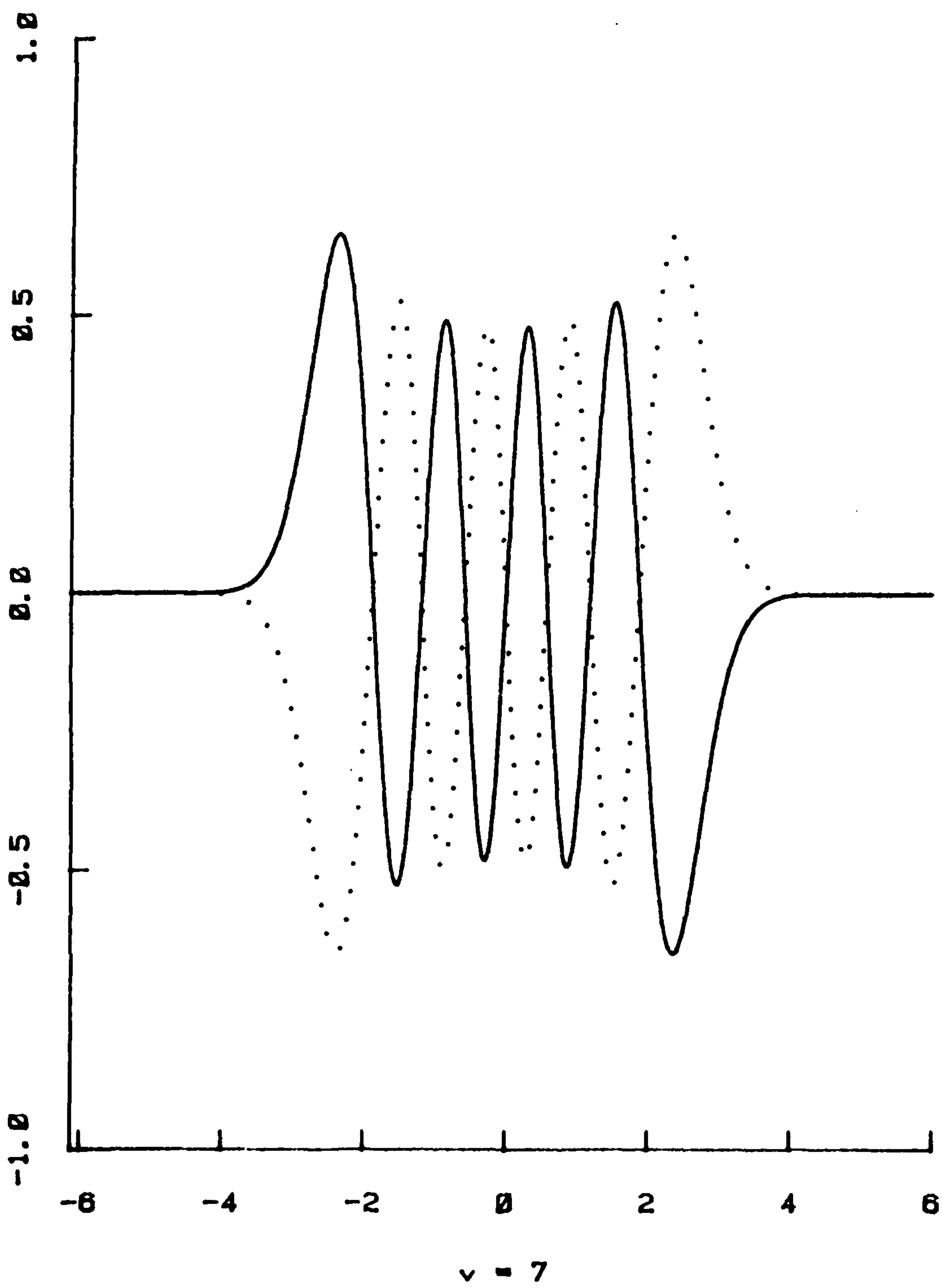


Fig. IV-1c. Selected Analytic and Numeric Wave Functions of the Single Harmonic Oscillator

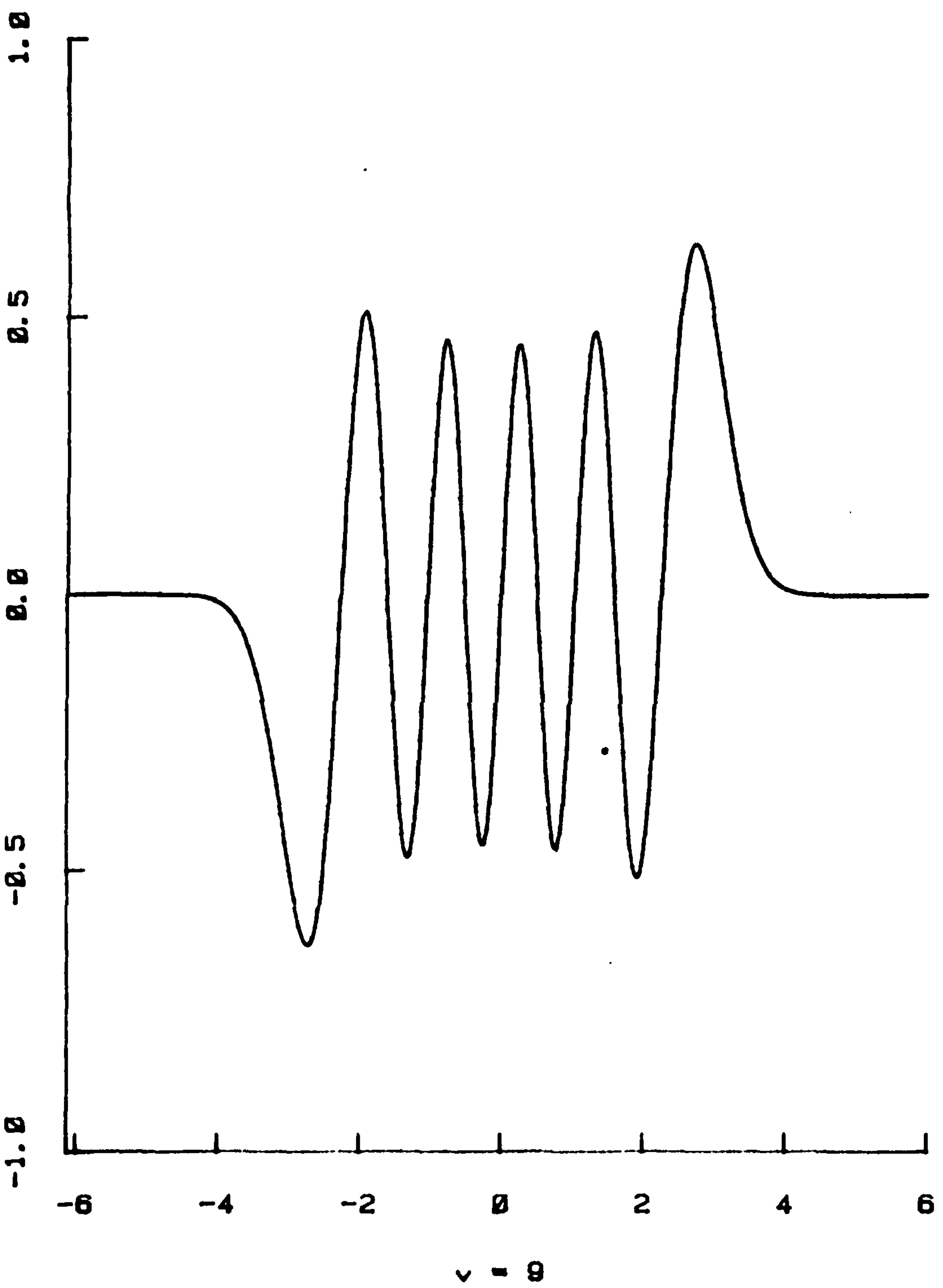


Fig. IV-1d. Selected Analytic and Numeric Wave Functions of the Single Harmonic Oscillator

Table IV-4

Test Runs of DIATOM Using MINUM
Correct Set P_1=2.0 P_2=0.0

RUN NUMBER	KEY WORDED RECORDS	RESULTS
1	NE=20 IS=100 P1=2.5 L1=0.1 U1=5.0 P2=0.0 L2=-10.0 U2=10.0	Chosen set: P_1=1.9863 P_2=0.0221 Sum of residuals squared = 1.6×10^{-3}
2	* P1=1.98 L1=1.90 U1=2.10 P2=0.02 L1=-0.02 U1=0.03	Chosen set: P_1=1.9800 P_2=0.0200 Sum of residuals squared = 7.8×10^{-3}
3	* NE=30	Chosen set: P_1=2.0061 P_2=-0.0196 Sum of residuals squared = 6.1×10^{-4}
4	* NE=40 IS=50 P1=2.000 L1=1.99 U1=2.01 P2=-0.01 L2=-0.02 U2=0.02	Chosen set: P_1=2.0004 P_2=-0.0019 Sum of residuals squared = 8.8×10^{-6}

* All other parameters the same as the previous run.

DIATOM appears to be capable of solving the wave equation accurately. The level of accuracy depends on the grid used. Also, DIATOM seems to be able to find the best set of potential energy parameters for the system of interest.

DIATOM is best used in a two step process. First, low resolution grids and large numbers of steps through MINUM are used to narrow the search region for each parameter. Then the resolution of the grid is increased while parameter limits are reduced until an acceptable fit of energy levels is achieved. Then MINUM is turned off and the number of grid elements increased so DIATOM can calculate accurate wave functions.

Validation of the Program FCFACT

The program FCFACT was tested only on the wave functions of the same harmonic oscillator problem. True Franck-Condon factors were not calculated since the wave functions used all came from the same potential. Therefore the factors computed by FCFACT were the square of the inner product $\langle \psi_i | \psi_j \rangle$ (i = 0 to 24; j = 0 to 24). For true Franck-Condon factors, ψ_i and ψ_j would belong to different potentials. Also, each set of wave functions would be calculated by runs of DIATOM.

The square of the inner product was as expected since the harmonic oscillator wave functions are orthogonal. If i = j then the inner product is 1.0. Otherwise it is zero.

These results were achieved with a grid of sixty elements. The square of the inner product was non-zero (.001 to .007) between high vibrational wave functions ($v > 15$) for a 45 element grid. These non-zero values occured only between ψ_i and ψ_j when $i = j \pm 2$.

The square of the inner product was also computed using Simpson's integration rule over 1001 points for $v = 0$ to 9. The spline coefficients were used to interpolate the 1001 points along the grid. All Simpson's rule values agreed with those of FCFACT.

The non-zero values arose since the numerical wave function only approximates the analytic wave function. The approximation could be made worse if a bad cubic spline fit is made to the numerical wave function. Still though, FCFACT gives good results for high resolution grids of 60 elements or more.

Program DIATOM Benchmark

Many runs of DIATOM were made in order to characterize its CPU time requirements. These runs are summarized in Table IV-5. The typical CPU requirements of the harmonic oscillator are presented. The first column indicates how many grid elements were used. Each grid began at $r = -6$ and ended at $r = 6$. The second column indicates how long the program took to solve the wave equation (RM=0, RW=1). MINUM was not allowed to run for this data. The third column

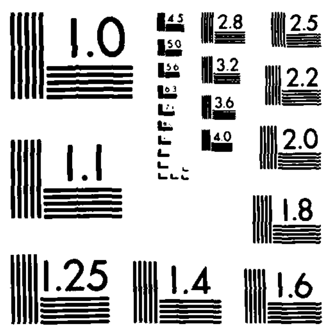
AD-A151 765 COMPUTER MODELING OF VIBRATIONAL ENERGY LEVELS OF POTENTIAL LASER CANDIDA..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. P H OSTDIEK 2/2

UNCLASSIFIED DEC 84 AFIT/GEP/PH/84D-6 F/G 20/8 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Table IV-5

An Overview of Program DIATOM's CPU Use

NUMBER OF GRID ELEMENTS	--RM=0, RW=1-- TOTAL CPU SEC	----RM=1, RW=0, IS=1---- TOTAL CPU SEC	ONE PASS CPU SEC	OVERHEAD CPU SEC
5	1.9	1.83	0.11	1.72
10	3.3	2.64	0.31	2.33
15	5.2	3.99	0.64	3.35
20	9.0	8.18	1.14	7.04
25	14.2	12.26	1.82	10.44
30	17.0	12.40	2.75	9.65
35	25.0	17.05	3.98	13.07
40	35.0	24.54	5.84	18.70
45	51.3	45.86	11.15	34.72
50	76.2	160.64	39.83	120.81
55	121.1	369.35	92.05	277.30
60	186.4	994.48	180.03	814.45

contains the CPU time required for DIATOM to run when MINUM was allowed only one step (RM=1, RW=0, IS=1). The fourth column contains the CPU time between calls of the function FUN by the subroutine MINUM. This is the approximate time for MINUM to take one step, and is referred to as the "one-pass" time. The fifth column contains the remainder of the CPU time for DIATOM to run. This is the overhead associated with the rest of the program and is referred to as the "overhead" time.

To approximate how long a run of DIATOM will take (RM=1, RW=0) use:

$$\text{CPU time} = (\text{overhead time}) + \text{IS}(\text{one-pass time}) \quad (83)$$

where IS is the number of steps MINUM is allowed to take. Eq (83) calculated the CPU time to within 30 seconds for the runs of DIATOM discussed previously.

V. Summary and Recommendations

Summary

Four programs were written to be used as a set to calculate the Franck-Condon factors between two electronic states of a diatomic molecule. The factors calculated are for v = 0 to 24. These programs are DUNHAM, EFIT, DIATOM, and FCFACT. DUNHAM calculates approximate energy levels using the Dunham equation (Eq (1)) and coefficients. Program EFIT uses a least squares technique to find a set of energy levels represented by spectroscopic data. Both programs are used to build the input file for DIATOM. This file contains the observed energy levels to which the MINUM portion of DIATOM will fit a potential energy curve. DIATOM uses a finite element technique to solve the wave equation and calculate wave function values. These wave function values are used by FCFACT to calculate Franck-Condon factors. DIATOM has derived very accurate wave functions for the single harmonic oscillator. Also, FCFACT has calculated the proper Franck-Condon factors between these wave functions. This program set promises to be an inexpensive and accurate alternative to the RKR-IPA program set. However, it still needs testing and modification.

Recommendations

1. Rewrite DIATOM to solve the wave equation in dimensionless form. This reduces the chance of mathematical operations exceeding storage limits (underflow/overflow errors) of the computer. Also by adding a small unit conversion subroutine, the user could use whatever units desired.

2. The execution time of DIATOM can be reduced by finding faster eigenvalue routines. Specifically, Dr. Shankland has written one routine which should be investigated, and used if faster than IMSL's EQRT1S.

3. DIATOM and FCFACT need to be tested against the analytic solutions of the Morse potential wave equation. This should be done as this work did for the single harmonic oscillator.

4. Compare the results of this program set and the RKR-IPA set for lead-oxide and lithium hydride. The RKR-IPA results are available in Pow's work (16).

5. Move the program set to the DEC VAX 11/780 and run under the UNIX operating system. This version of the programs would be more transportable since both DEC equipment and the UNIX operating system are popular in laboratories.

Bibliography

1. Bowdler, H., Martin, R.S., Reinsch, C., and Wilkinson, J.H. "The QR and QL Algorithm for Symmetric Matricies," Numerische Mathematik 11 (4):293-306 (1968).

2. De Boor, C. A Practical Guide to Splines. New York: Springer-Verlag, 1978.

3. Desai, C.S. and Abel, J.F. Introduction to the Finite Element Method. New York: Van Nostrand Reinhold Co, 1972.

4. Dongana, J.J., Moler, C.B., Birch, J.R., and Stewart, G.W. LINPACK User's Guide. Philadelphia: SIAM, 1979.

5. Dorko, E.A. "Calculation of Potential Energy Curves and Franck-Condon Factors for Diatomic Molecules by Use of the Inverted Approach", 39th Symposium on Molecular Spectroscopy, Ohio State University, Columbus, Ohio, June 11, 1984.

6. Dunham, J.L. "The Energy Levels of a Rotating Vibrator," Physical Review 41: 721 (1932).

7. French, A.P., and Taylor, E.F. An Introduction to Quantum Mechanics. New York: W.W. Norton and Company, Inc., 1978.

8. Gouedard, G. and Vique, J. "Improvement of the RKR Procedure Obtained by Semi-Classical Calculations," Chemical Physics Letters, 96: 293 (1983).

9. Harris Corporation. Harris Fortran Reference Manual. 2101 Cypress Creek Rd. Fort Lauderdale, Florida, 1980.

10. Herzberg, G. Spectra of Diatomic Molecules. New York: McGraw Hill, 1950.

11. IMSL. IMSL Library Reference Manual. 7500 Bellaire Blvd., Houston, Texas, 1982.

12. Kirschner, S.M. and Watson, J.K.G. "Second-Order Semiclassical Calculations for Diatomic Molecules," Journal of Molecular Spectroscopy, 51: 321 (1974).

13. Martin, R.S., Reinsch, C. and Wilkinson, J.H. "Householders Tridagonalization of a Symmetric Matrix," Numerische Mathematik 11 (3): 181-195 (1968).

14. Moelwyn-Hughes, E.A. Physical Chemistry. New York: Pergamon Press Ltd, 1964.

15. Pearson, C.E. and Willman, W.S. "A Computer Program for Design Optimization," Sperry Rand Research Center Report No. SRRC-RR-65-48, June 1965.

16. Pow, J.J. The Development of Computer Routines to Perform a Comprehensive Analysis of Spectroscopic Data from Diatomic Molecules. MS Thesis. Air Force Institute of Technology, Wright-Patterson Air Force Base, Oh, December 1983.

17. Rao, S.S. The Finite Element Method in Engineering. New York: Pergamon Press, 1982.

18. Reinsch, C. and Bauer, F.L. "Rational QR Transformation with Newton Shift for Symmetric Tridiagonal Matricies," Numerische Mathematik 11 (3): 264-272 (1968).

19. Richards, W.G. and Barrow, R.F. "The Calculation of Potential Curves for Diatomic Molecules From Experimental Data," Proc. Phys. Soc. (London) 83: 1045 (1964).

20. Rutger, L.L. Numerical Methods for the Preparation of Potential Energy Curves of Diatomic Molecules. MS Thesis. Air Force Institute of Technology, wright-Patterson Air Force Base, Oh, March 1983.

21. Shankland, D. Private Lecture Notes, Thesis Preparation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson Air Force Base, Oh, May - September 1984.

22. Steinfeld, Jeffrey I. Molecules and Radiation, An Introduction to Modern Molecular Spectroscopy. Cambridge: MIT Press, 1974.

23. Sur, A. and Tellinghuisen, J. "The D'--A' Transition in Br_2," Journal of Molecular Spectroscopy 88: 323 (1981).

24. Tellinghuisen, J. and Henderson, A.D. "The Use of Morse -RKR Curves in Diatomic Calculations," Chemical Physics Letters 91: 447 (1982).

25. Vanderslice, J.T., Mason, E.A. and Maisch, W.G. "Interactions between Ground State Oxygen Atoms and Molecules: O-O and O_2-O_2," Journal of Chemical Physics 32: 515 (1960).

26. Vidal C.R. and Scheingraber, H. "Determination of Diatomic Molecular Constants Using an Inverted Perturbation Approach," Journal of Molecular spectroscopy 65: 46 (1977).

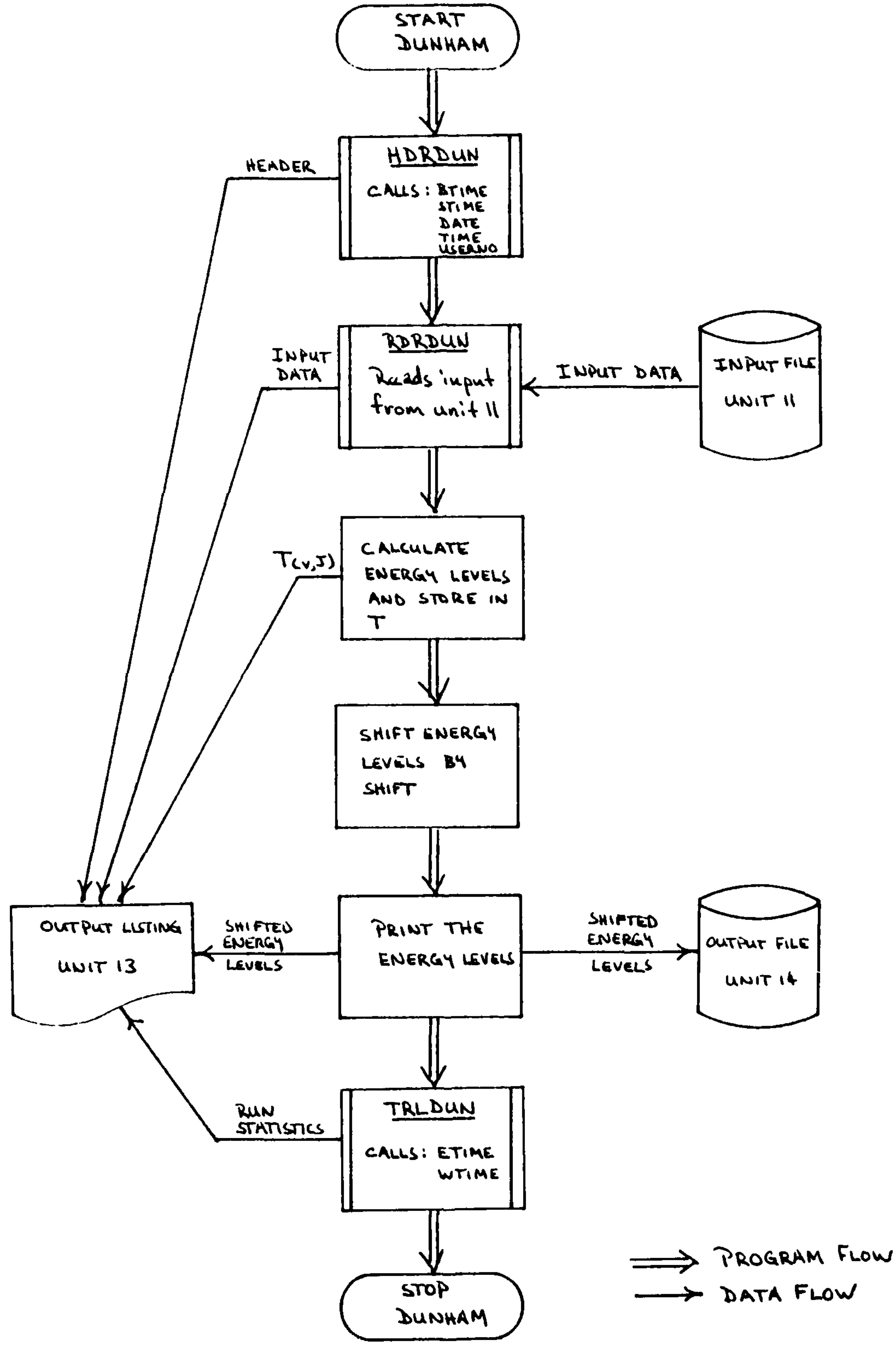
27. Wells, B.H., Smith, B. and Zare, R.N. "The Stability of the RKR Inversion Procedure to Errors in the Spectroscopic Data: Origin of the Inner Wall Ripple," Chemical Physics Letters 99: 244 (1983).

28. Wilkinson, J.H. The Algebraic Eigenvalue Problem. Oxford: Clarendon Press, 1965.

29. Yang, S.C. "The Avoided Crossing Region of the CsH X^1 + Potential Energy Curve," Journal of Chemical Physics 77: 2884 (1982).

Appendix A

Program DUNHAM Flow



Appendix B

Program DUNHAM

```
C-------------------------------------------------------------------------
C
C      Program:   DUNHAM
C
C      Version:   84.06.13
C
C       Author:   Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This program calculates approximate energy levels
C                    for diatomic molecules using the Dunham equation.
C
C                    I/O logical unit  6 -- output listing file
C                                     11 -- input file
C                                     13 -- output listing file
C                                           (same as unit 6)
C                                     14 -- energy level output file
C
C-------------------------------------------------------------------------

      PROGRAM MAIN

      CHARACTER*2  NUMBER(0:25)
      CHARACTER*30 HEADER

      INTEGER      I, J, LVLLMT, VIB, VIBLMT, ROT, ROTLMT, EVIB(676),
     +             EROT(676)

      REAL         SUM, T(0:25,0:25), Y(0:9,0:9), RROT(0:25),
     +             RVIB(0:25), ELVL(676), DEQUIL

      COMMON /HDR/ HEADER
      COMMON /DAT/ Y, DEQUIL, VIBLMT, ROTLMT, LVLLMT

      DATA NUMBER /'00','01','02','03','04','05','06','07','08','09',
     +             '10','11','12','13','14','15','16','17','18','19',
     +             '20','21','22','23','24','25'/

      DATA RROT / 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0,
     +           10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,18.0,19.0,
     +           20.0,21.0,22.0,23.0,24.0,25.0/

      DATA RVIB / 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0,
     +           10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,18.0,19.0,
     +           20.0,21.0,22.0,23.0,24.0,25.0/

C     --------------------------------Open the output listing file, print
C                                     the header, and start run stats.

      CALL HDRDUN
```

```
C     ------------------------------Read the input file (unit 11) for
C                                   the control parameters and data.

      CALL RDRDUN

      WRITE (13,1303)
1303  FORMAT ('1 Energies T(v,J)  (T=0 at potential minimum)',/)

C     ------------------------------Find the Dunnam Equation energy
C                                   T(VIB,ROT) where:
C                                        VIB is the vibrational quantum
C                                            number of the current level
C                                        ROT is the rotational quantum
C                                            number of the current level

      DO 40 VIB=0,VIBLMT
        DO 30 ROT=0,ROTLMT
          SUM = 0

C         --------------------------The sum (do loop) indicies may vary
C                                   from 0 to 9.  This yields the first
C                                   100 terms of the Dunham Equation.

          DO 20 I=0,9
            DO 10 J=0,9
              IF (I .EQ. 0 .AND. J .EQ. 0) THEN
                SUM = SUM + Y(I,J)
              ELSE
                IF (I .EQ. 0 .AND. J .NE. 0) THEN
                  SUM = SUM+(Y(I,J)*RROT(ROT)**J*(RROT(ROT)+1.0)**J)
                ELSE
                  IF (I .NE. 0 .AND. J .EQ. 0) THEN
                    SUM = SUM + (Y(I,J) * (RVIB(VIB) + 0.5)**I)
                  ELSE
                    SUM = SUM+(Y(I,J)*(RVIB(VIB)+0.5)**I
     +                    *RROT(ROT)**J*(RROT(ROT)+1.0)**J)
                  ENDIF
                ENDIF
              ENDIF
10          CONTINUE
20        CONTINUE

C         --------------------------Transfer the accumulated sum to T

          T(VIB,ROT) = SUM
          WRITE (13,1304) VIB, ROT, T(VIB,ROT)
1304      FORMAT (' T(',I2,',',I2,')=',G15.7)
30      CONTINUE
40    CONTINUE

C     ------------------------------Transfer the calculated energy
C                                   levels T(VIB,ROT) to ELVL(I).
```

```
      I = 0
      DO 60 VIB=0,VIBLMT
        DO 50 ROT=0,ROTLMT
          I = I + 1
          ELVL(I) = T(VIB,ROT)
          EVIB(I) = VIB
          EROT(I) = ROT
50      CONTINUE
60    CONTINUE

C     -------------------------------Shift the energy levels so that the
C                                    lowest (VIB=0 & ROT=0) is at
C                                    0 - (the dissociation energy)

      DO 70 I=1,LVLLMT
        ELVL(I) = ELVL(I) - DEQUIL
70    CONTINUE

      WRITE (13,1301) HEADER
1301  FORMAT ('1Dunham Equation Energy Levels  ',A30,/)

      DO 80 I=1,LVLLMT
        WRITE (13,1302) EVIB(I), EROT(I), ELVL(I)
1302    FORMAT (' V=',I4,' J=',I4,1X,G15.7)
80    CONTINUE

C     -------------------------------Put the energy levels into an
C                                    output file (unit 14).

      OPEN (UNIT=14)

      DO 90 I=1,LVLLMT
        WRITE (14,1401) NUMBER(I-1), ELVL(I)
1401    FORMAT ('?V',A2,'=',G15.7,' ; ')
90    CONTINUE

      CLOSE (UNIT=14)

C     -------------------------------Close the output file and run
C                                    statistics.

      CALL TRLDUN

9999  END
$ADD,HDRDUN
$ADD,RDRDUN
$ADD,TRLDUN
```

```
C-----------------------------------------------------------------------
C
C      Program:  SUBROUTINE HDRDUN
C
C      Version:  84.08.13
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  HDRDUN opens the output listing file (logical units
C                    13 and 6) and starts CPU and wall time use
C                    statistics.
C
C-----------------------------------------------------------------------

      SUBROUTINE HDRDUN

      CHARACTER*8 VERSN
      CHARACTER*13 PCN

      INTEGER*3 IDATE(3), ITIME(3), IUSER(4)

      VERSN = '84.08.13'
      PCN   = 'GEP/84D-6/1.2'

C      ------------------------------Initiate the run statistics.

      CALL BTIME
      CALL STIME

C      ------------------------------Get the current date, time, and
C                                    user name for output on the header

      CALL DATE(IDATE)

      CALL TIME(ITIME)

      CALL USERNO(IUSER)

C      ------------------------------Open the output listing file and
C                                    write out the header

      OPEN (UNIT=13)
      OPEN (UNIT=6)

      WRITE (13,1301) IUSER, VERSN, IDATE, PCN, ITIME
 1301 FORMAT ('1 User:  ',4A3,T51,'Air Force Institute of Technology',
     +          T110,'Version:  ',A8,
     +        /,'  Date: ',3A3,T114,'PCN:  ',A13,
     +        /,'  Time: ',3A3,T54,'DUNHAM',/,/)

 9999 RETURN
      END
```

```
            ENDIF
220       CONTINUE
        ELSE
          IF (SCORE(2:5) .EQ. 'LVLS') THEN

C           ----------------------Find the correct state number xx of
C                                 LVLSxx where xx is '01' to '10' and
C                                 corresponds to STATxx above.

            DO 250 I=1,10
              IF (SCORE(6:7) .EQ. DIGITS(I)) THEN

C               ------------------Count how many levels are associated
C                                 with the state xx and record them in
C                                 STATE(xx,count)

                SCOL = 9
                DO 240 J=1,10
                  DO 230 L=SCOL,71
                    IF (SCORE(L:L) .NE. ' ') THEN
                      IF (NINST(I) .LT. 25) THEN
                        NINST(I) = NINST(I) + 1
                        READ (SCORE(L:(L+1)),'(I2)')
     +                       STATE(I,NINST(I))
                        SCOL = L + 2
                      ENDIF
                    ENDIF
230               CONTINUE
240             CONTINUE
                GO TO 80
              ENDIF
250         CONTINUE
          ELSE
            IF (SCORE(2:7) .EQ. 'SHIFTS') THEN

C             --------------------SHIFT is the value desired for the
C                                 lowest level of the lowest state.

              READ (SCORE(9:23),'(E15.7)') SHIFT
            ELSE

C             --------------------Look for a keyword ABBCDD where:
C                                  A & C  match values in STLBL
C                                 BB & DD match values in STATE.

              FOUND = .FALSE.

C             --------------------Look for part A.

              DO 310 I=1,10
                IF (SCORE(2:2) .EQ. STLBL(I)) THEN
                  CURST(1:1) = SCORE(2:2)
                  ILBL       = I
                ENDIF
310           CONTINUE
```

```
      DO 3 I=1,10
        STLBL(I) = ' '
        NINST(I) = 0
        DO 2 J=1,25
          STATE(I,J) = 0
2       CONTINUE
3     CONTINUE

      DO 5 I=1,250
        DO 4 J=1,250
          LINE(I,J) = 0
          WEIGHT(I,J) = 0
4       CONTINUE
5     CONTINUE

C     ------------------------------Open the input data file.

      OPEN (UNIT=11)

      WRITE (13,1301)
1301  FORMAT (/,'0',T31,'Input Data',/,' ',T31,'----- ----')

      DO 80 K=1,10000

C       ----------------------------Transfer a record from the input
C                                   file to the buffer SCORE.

        READ (11,1101,END=81) SCORE
1101    FORMAT (A72)

        WRITE (13,1302) SCORE
1302    FORMAT (' ',A72)

C       ----------------------------If SCORE(1:1) is a '>', then SCORE
C                                   should contain data and a valid
C                                   keyword.  So, see if SCORE(2:7)
C                                   does contain a valid keyword.  If it
C                                   does, transfer the data from SCORE
C                                   to the input variable.

        IF (SCORE(1:1) .EQ. '>') THEN
          IF (SCORE(2:5) .EQ. 'STAT') THEN

C           ------------------------Find the correct state number STATxx
C                                   where xx is '01' to '10'.

            DO 220 I=1,10
              IF (SCORE(6:7) .EQ. DIGITS(I)) THEN
                DO 210 J=9,72
                  IF (SCORE(J:J) .NE. ' ') THEN
                    STLBL(I)(1:1) = SCORE(J:J)
                    SETCNT        = SETCNT + 1
                    GO TO 80
                  ENDIF
210             CONTINUE
```

```
C-------------------------------------------------------------------
C
C     Program:  SUBROUTINE RDREFT
C
C     Version:  84.08.13
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine opens an input file (unit 11) and reads
C                   all records within.  Each record read is written to
C                   the output listing (unit 13).  Data records are
C                   marked by a '>' in column 1.  These records contain
C                   data referenced by a single key word.  All other
C                   records are considered comments.  This routine uses
C                   an internal read, eg READ (SCORE(5:19),'(E15.7)') X
C                   reads from columns 5 to 19 of the character
C                   variable SCORE using the edit descriptor E15.7 into
C                   the real variable X.
C
C-------------------------------------------------------------------

      SUBROUTINE RDREFT

      CHARACTER*1  CURST, STLBL(10)
      CHARACTER*2  NUMB26(26), DIGITS(10)
      CHARACTER*7  FORM
      CHARACTER*72 SCORE

      INTEGER I, L1, J, L2, K, SIZE, FCOL, LCOL, NINST(10),
     +        CURLVL, LVLNO, ILBL, SETCNT, STATE(10,25)

      LOGICAL FOUND

      REAL    LINE(250,250), WEIGHT(250,250), SHIFT

      COMMON /INDATA/ LINE, WEIGHT, SIZE, SHIFT
      COMMON /LABELS/ STLBL, CURST
      COMMON /LEVELS/ NINST, STATE, CURLVL, SETCNT

      DATA DIGITS /'01','02','03','04','05','06','07','08','09','10'/

      DATA NUMB26 /'00','01','02','03','04','05','06','07','08','09',
     +             '10','11','12','13','14','15','16','17','18','19',
     +             '20','21','22','23','24','25'/

C     ------------------------------Initialize the input variables.

      SIZE = 0
      SETCNT = 0
```

```
C-----------------------------------------------------------------------
C
C      Program:  SUBROUTINE HDREFT
C
C      Version:  84.08.13
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  HDREFT opens the output listing file (logical units
C                    13 and 6) and starts CPU and wall time use
C                    statistics.
C
C-----------------------------------------------------------------------

      SUBROUTINE HDREFT

      CHARACTER*8 VERSN
      CHARACTER*13 PCN

      INTEGER*3 IDATE(3), ITIME(3), IUSER(4)

      VERSN = '84.08.13'
      PCN   = 'GEP/84D-6/1.1'

C     ------------------------------Initiate the run statistics.

      CALL ETIME
      CALL STIME

C     ------------------------------Get the current date, time, and
C                                   user name for output on the header

      CALL DATE(IDATE)

      CALL TIME(ITIME)

      CALL USERNO(IUSER)

C     ------------------------------Open the output listing file and
C                                   write out the header

      OPEN (UNIT=13)
      OPEN (UNIT=6)

      WRITE (13,1301) IUSER, VERSN, IDATE, PCN, ITIME
1301  FORMAT ('1 User:  ',4A3,T51,'Air Force Institute of Technology',
     +          T110,'Version:  ',A8,
     +       /,'  Date: ',3A3,T114,'PCN:  ',A13,
     +       /,'  Time: ',3A3,T62,'ENERGY FIT',/,/)

9999  RETURN
      END
```

```
      DO 190 I=1,SIZE
        X(I) = X(I) + SHIFT
190   CONTINUE

      WRITE (13,1303) SHIFT
1303  FORMAT ('1Energy levels shifted so lowest level is ',G15.7)
      WRITE (13,1302) (X(I),I=1,SIZE)

C     ------------------------------Write the energy levels to the
C                                   output listing (unit 13) and an
C                                   output file (unit 14).

      OPEN (UNIT=14)

      OFFSET = 0

      WRITE (13,1312)
1312  FORMAT ('1The energy levels for each state are:')

        DO 400 I=1,SETCNT
          WRITE (13,1310) STLBL(I)
1310      FORMAT ('0Levels of state: ',A1)
          DO 410 J=1,NINST(I)
            WRITE (13,1311) STATE(I,J), X(OFFSET+J)
1311        FORMAT (' ',I6,G15.7)
            WRITE (14,1401) NUMBER(STATE(I,J)), X(OFFSET+J)
1401        FORMAT ('>V',A2,'=',G15.7,' ; ')
410       CONTINUE
        OFFSET = OFFSET + NINST(I)
400     CONTINUE

      CLOSE (UNIT=14)

C     ------------------------------Close the output file and run
C                                   statistics.

      CALL TRLEFT

9999  END
$ADD,RDREFT
$ADD,LVLEFT
$ADD,HDREFT
$ADD,TRLEFT
```

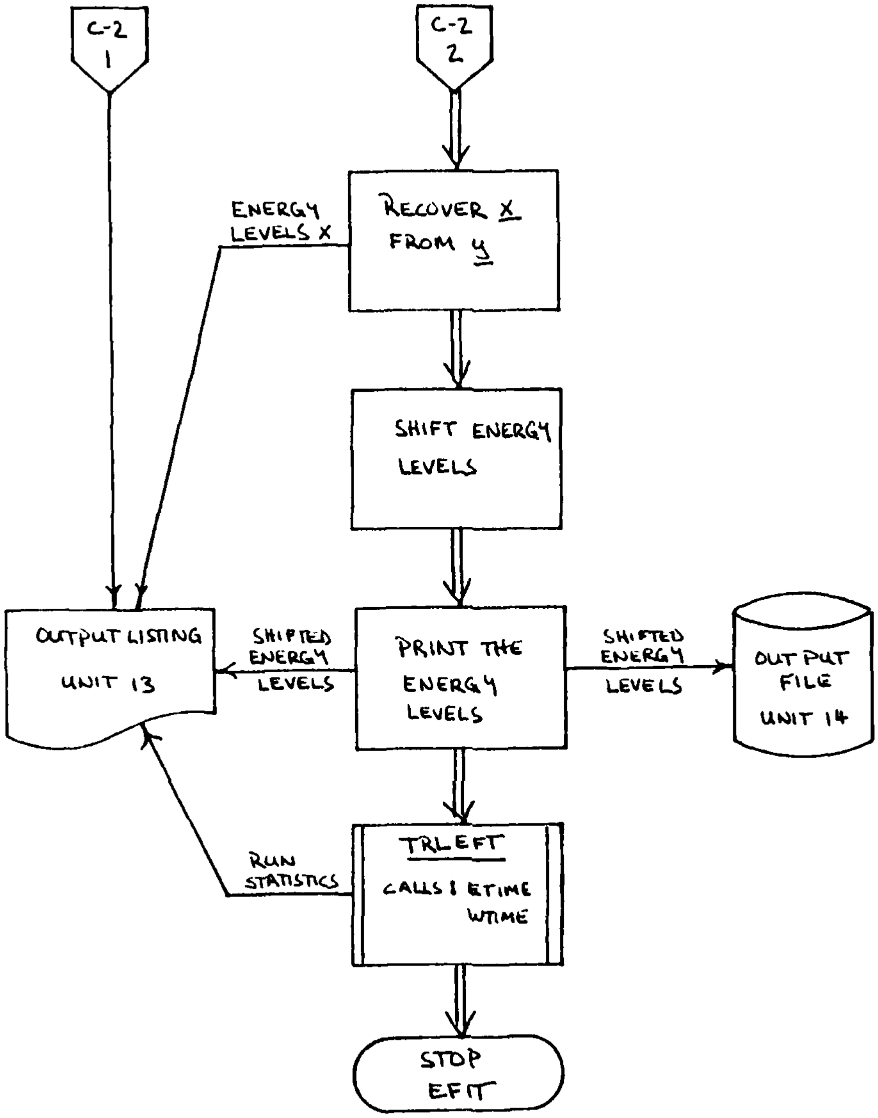
```
      IF (I .EQ. J) THEN
        A(I,J) = SQRT(A(I,J) - SUM)
      ELSE
        A(I,J) = (A(I,J) - SUM) / A(J,J)
      ENDIF
110     CONTINUE
120   CONTINUE

C     ------------------------------Solve the equation:
C                                               LY = B
C                                   where Y = L(transpose) x X
C                                   and is stored in X.

      DO 140 I=1,(SIZE-1)
        SUM = 0
        DO 130 K=1,(I-1)
          SUM = SUM + (A(I,K) * X(K))
130     CONTINUE
        X(I) = (B(I) - SUM) / A(I,I)
140   CONTINUE
      X(SIZE) = 0

C     ------------------------------This is a consistency check.

      SUM = 0
      DO 150 K=1,(SIZE-1)
        SUM = SUM + (A(SIZE,K) * X(K))
150   CONTINUE
      WRITE (13,1304)
1304  FORMAT ('1Consistency check...')
      WRITE (13,1301) SUM, SIZE, B(SIZE)
1301  FORMAT ('   SUM = ',G15.7,/,' B(' ,I2,') = ',G15.7)

C     ------------------------------Recover X from Y (stored in X).

      DO 170 I=(SIZE-1),1,-1
        SUM = 0
        DO 160 K=(I+1),(SIZE-1),1
          SUM = SUM + (A(K,I) * X(K))
160     CONTINUE
        X(I) = (X(I) - SUM) / A(I,I)
170   CONTINUE


C     ------------------------------Write out X

      WRITE (13,1305)
1305  FORMAT (/,' Unshifted energy levels are...')
      WRITE (13,1302) (X(I),I=1,SIZE)
1302  FORMAT ('>',G15.7)

C     ------------------------------Shift the energy values.

      DO 180 I=SIZE,1,-1
        X(I) = X(I) - X(1)
180   CONTINUE
```

```
C       ------------------------------Build A's diagonals of AX=B.

        DO 30 K=1,SIZE
          SUMIN = 0
          DO 10 I=1,SIZE
            SUMIN = SUMIN + WEIGHT(I,K)
10        CONTINUE
          SUMOUT = 0
          DO 20 I=1,SIZE
            SUMOUT = SUMOUT + WEIGHT(K,I)
20        CONTINUE
          A(K,K) = SUMIN + SUMOUT
30      CONTINUE

C       ------------------------------Build non-diagonals of A.

        DO 60 K=1,SIZE
          DO 50 J=1,(K-1)
            A(K,J) = -1.0 * (WEIGHT(K,J) + WEIGHT(J,K))
            A(J,K) = A(K,J)
50        CONTINUE
60      CONTINUE


C       ------------------------------Build B

        DO 90 K=1,SIZE
          SUMOUT = 0
          DO 70 J=1,SIZE
            SUMOUT = SUMOUT + (WEIGHT(K,J) * LINE(K,J))
70        CONTINUE
          SUMIN = 0
          DO 80 J=1,SIZE
            SUMIN = SUMIN + (WEIGHT(J,K) * LINE(J,K))
80        CONTINUE
          B(K) = SUMOUT - SUMIN
90      CONTINUE

C       ------------------------------Solve for X
C
C                                    First use Cholesky decomposition to
C                                    get L (stored in A) such that:
C                                             A = L x L(transpose).

        DO 120 J=1,SIZE
          DO 110 I=J,SIZE
            SUM = 0
            DO 100 K=1,(J-1)
              IF (I .EQ. J) THEN
                SUM - SUM + (A(I,K)**2)
              ELSE
                SUM = SUM + (A(I,K) * A(J,K))
              ENDIF
100         CONTINUE
```

Appendix D

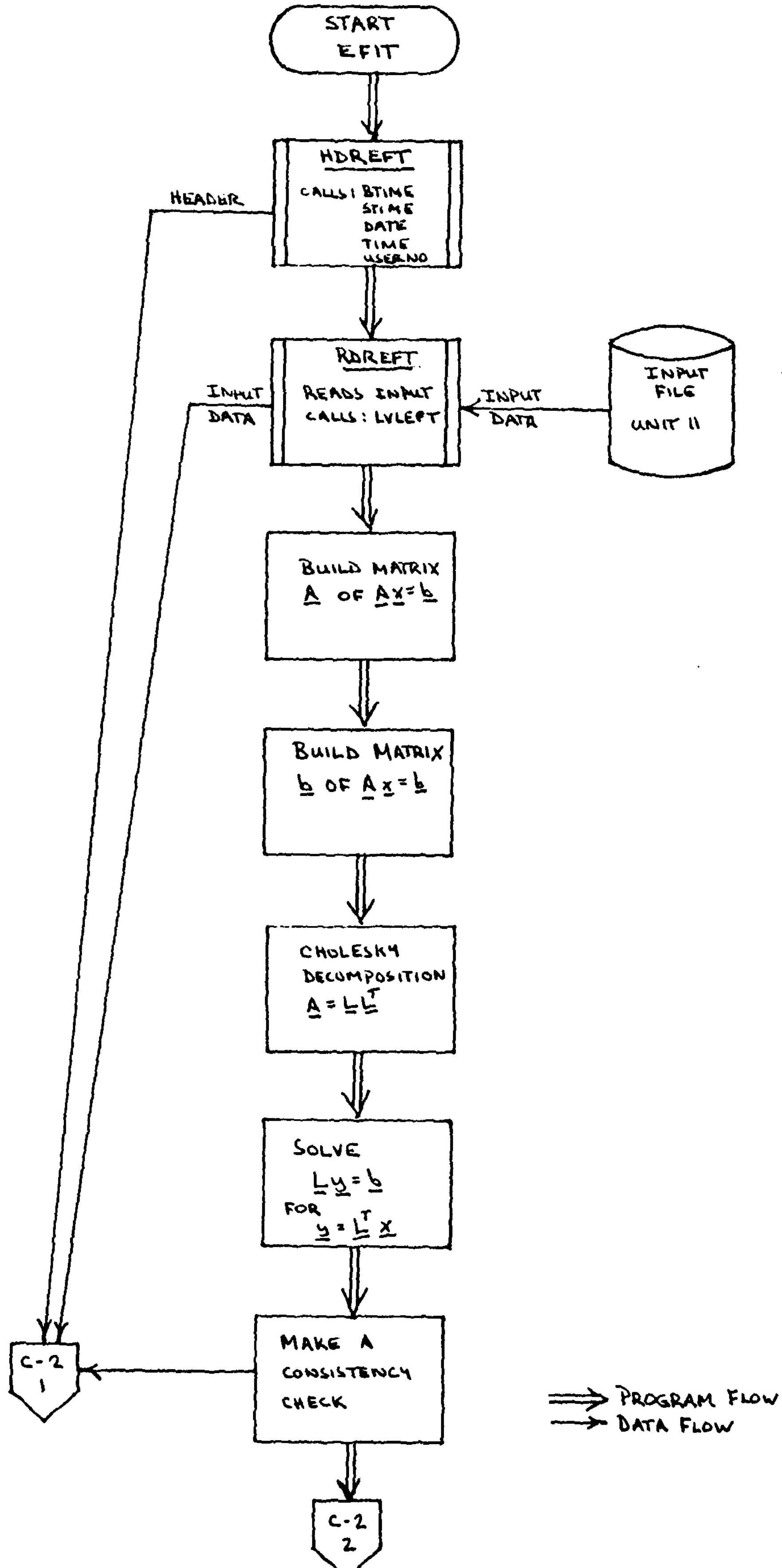
Program EFIT

```
C-----------------------------------------------------------------------
C
C     Program:   EFIT
C
C     Version:   84.08.13
C
C      Author:   Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This program uses a least squares technique to find
C                   the set of energy levels that best fit a set of
C                   spectroscopic transition lines.
C
C                   I/O logical unit  6 -- output listing file
C                                    11 -- input file
C                                    13 -- output listing file
C                                          (same as unit 6)
C                                    14 -- energy level output file
C
C-----------------------------------------------------------------------

      PROGRAM MAIN

      CHARACTER*1 CURST, STLBL(10)
      CHARACTER*2  NUMBER(0:25)

      INTEGER I, J, K, SIZE, NINST(10), STATE(10,25), CURLVL,
     +        SETCNT, OFFSET

      REAL A(250,250), X(250), B(250), SUM, SHIFT, LINE(250,250),
     +     WEIGHT(250,250), SUMIN, SUMOUT

      COMMON A
      COMMON /INDATA/ LINE, WEIGHT, SIZE, SHIFT
      COMMON /LABELS/ STLBL, CURST
      COMMON /LEVELS/ NINST, STATE, CURLVL, SETCNT

      DATA NUMBER /'00','01','02','03','04','05','06','07','08','09',
     +            '10','11','12','13','14','15','16','17','18','19',
     +            '20','21','22','23','24','25'/

C     ------------------------------Open the output listing file, print
C                                   the header, and start run stats.

      CALL HDREFT

C     ------------------------------Read in measured line values and
C                                   control parameters.

      CALL RDREFT
```


C-2
1
C-2
2
RECOVER X FROM y
ENERGY LEVELS X
SHIFT ENERGY LEVELS
OUTPUT LISTING
UNIT 13
SHIFTED ENERGY LEVELS
PRINT THE ENERGY LEVELS
SHIFTED ENERGY LEVELS
OUT PUT FILE
UNIT 14
RUN STATISTICS
TRLEFT
CALLS ETIME WTIME
STOP
EFIT

Appendix C

Program EFIT Flow



```
C-------------------------------------------------------------------
C
C      Program:  SUBROUTINE TRLDUN
C
C      Version:  84.08.13
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  TRLDUN closes the output listing file (logical
C                    units 13 and 6) and stops the CPU and wall time
C                    use statistics.
C
C-------------------------------------------------------------------

      SUBROUTINE TRLDUN

C     ------------------------------Shut down the run statistics.

      CALL ETIME
      CALL WTIME

C     ------------------------------Close the listing outputs.

      CLOSE (UNIT=13,STATUS='KEEP')
      CLOSE (UNIT=6)

9999  RETURN
      END
```

```
                  READ (SCORE(6:20),'(E15.7)') Y(II,JJ)
                 ENDIF
                ENDIF
               ENDIF
              ENDIF
             ENDIF
            ENDIF
           ENDIF
30    CONTINUE

C     ------------------------------Close the input file.

31    CLOSE (UNIT=11)

9999  RETURN
      END
```

```
C       ------------------------------Open the input data file.

        OPEN (UNIT=11)

        DO 30 K=1,1000

C          ----------------------------Transfer a record from the input
C                                      file to the buffer SCORE.

           READ (11,1101,END=31) SCORE
1101       FORMAT (A72)

           WRITE (13,1302) SCORE
1302       FORMAT (' ',A72)

C          ----------------------------If SCORE(1:1) is a '>', then SCORE
C                                      should contain data and a valid
C                                      keyword.  So, see if SCORE(2:4)
C                                      does contain a valid keyword.  If it
C                                      does, transfer the data from SCORE
C                                      to the input variable.

           IF (SCORE(1:1) .EQ. '>') THEN
             IF (SCORE(2:4) .EQ. 'VIB') THEN
               READ (SCORE(6:20),'(I15)') VIBLMT
             ELSE
               IF (SCORE(2:4) .EQ. 'ROT') THEN
                 READ (SCORE(6:20),'(I15)') ROTLMT
               ELSE
                 IF (SCORE(2:4) .EQ. 'LVL') THEN
                   READ (SCORE(6:20),'(I15)') LVLLMT
                 ELSE
                   IF (SCORE(2:4) .EQ. 'HDR') THEN
                     HEADER(1:30) = SCORE(6:35)
                   ELSE
                     IF (SCORE(2:4) .EQ. 'DEQ') THEN
                       READ (SCORE(6:20),'(E15.7)') DEQUIL
                     ELSE

C                        ----------------If SCORE(2:2) is a 'Y', then SCORE
C                                      may contain a Dunham Coefficient.
C                                      Use DIGIT to find which one it is.

                         IF (SCORE(2:2) .EQ. 'Y') THEN
                           DO 10 I=0,9
                             IF (SCORE(3:3) .EQ. DIGIT(I)) THEN
                               II = I
                             ENDIF
10                         CONTINUE
                           DO 20 I=0,9
                             IF (SCORE(4:4) .EQ. DIGIT(I)) THEN
                               JJ = I
                             ENDIF
20                         CONTINUE
```

```
C --------------------------------------------------------------------
C
C      Program:  SUBROUTINE RDRDUN
C
C      Version:  84.08.13
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine opens an input file (unit 11) and reads
C                    all records within.  Each record read is written to
C                    the output listing (unit 13).  Data records are
C                    marked by a '>' in column 1.  These records contain
C                    data referenced by a single key word.  All other
C                    records are considered comments.  This routine uses
C                    an internal read, eg READ (SCORE(5:19),'(E15.7)') X
C                    reads from columns 5 to 19 of the character
C                    variable SCORE using the edit descriptor E15.7 into
C                    the real variable X.
C
C --------------------------------------------------------------------

      SUBROUTINE RDRDUN

      CHARACTER*1  DIGIT(0:9)
      CHARACTER*30 HEADER
      CHARACTER*72 SCORE

      INTEGER      I, II, J, JJ, LVLLMT, ROTLMT, VIBLMT

      REAL         Y(0:9,0:9), DEQUIL

      COMMON /HDR/ HEADER
      COMMON /DAT/ Y, DEQUIL, VIBLMT, ROTLMT, LVLLMT

      DATA DIGIT /'0','1','2','3','4','5','6','7','8','9'/

      WRITE (13,1301)
 1301 FORMAT (/,'0',T31,'Input Data',/,' ',T31,'----- ----')

C     ------------------------------Initialize the input variables.

      VIBLMT = 0
      ROTLMT = 0
      LVLLMT = 0

      DO 5 I=0,9
        DO 4 J=0,9
          Y(I,J) = 0
    4   CONTINUE
    5 CONTINUE
```

```
C                     --------------------Look for part BB.

                      DO 20 I=1,26
                        IF (SCORE(3:4) .EQ. NUMB26(I)) THEN
                          DO 320 J=1,25
                            IF ((I-1) .EQ. STATE(ILBL,J)) THEN
                              CURLVL = I - 1

C                             ------------Subroutine LVLEFT returns the
C                                         absolute level number (LVLNO) for a
C                                         for a given level relative to a
C                                         given electronic state.

                              CALL LVLEFT (LVLNO)

                              L1 = LVLNO
                            ENDIF
320                       CONTINUE
                          IF (L1 .GT. SIZE) SIZE = L1
                          GO TO 21
                        ENDIF
20                    CONTINUE
                      GO TO 31

C                     --------------------Look for part C.

21                    DO 330 I=1,10
                        IF (SCORE(5:5) .EQ. STLBL(I)) THEN
                          CURST(1:1) = SCORE(5:5)
                          ILBL       = I
                        ENDIF
330                   CONTINUE

C                     --------------------Look for part DD.

                      DO 30 I=1,26
                        IF (SCORE(6:7) .EQ. NUMB26(I)) THEN
                          DO 340 J=1,25
                            IF ((I-1) .EQ. STATE(ILBL,J)) THEN
                              CURLVL = I - 1

C                             ------------Subroutine LVLEFT returns the
C                                         absolute level number (LVLNO) for a
C                                         for a given level relative to a
C                                         given electronic state.

                              CALL LVLEFT (LVLNO)

                              L2 = LVLNO
                            ENDIF
340                       CONTINUE
                          IF (L2 .GT. SIZE) SIZE = L2
                          FOUND = .TRUE.
                          GO TO 31
                        ENDIF
30                    CONTINUE
```

```
C                   --------------------If a correct keyword has been found
C                                       for a transition from state 'A',
C                                       level 'BB' to state 'C', level 'DD'.
C                                       Now separate the value for the
C                                       transition line from its weighting
C                                       factor.  They are separated by a ';'

31                  IF (FOUND) THEN
                      FCOL = 0
                      LCOL = 0
                      FORM = '(E15.7)'
                      DO 40 J=9,72
                        IF (SCORE(J:J) .NE. ' ' .AND.
     +                      SCORE(J:J) .NE. ';') THEN
                          FCOL = J
                          GO TO 41
                        ENDIF
40                    CONTINUE
41                    IF (FCOL .GT. 0) THEN
                        DO 50 J=FCOL,72
                          IF (SCORE(J:J) .EQ. ' ' .OR.
     +                        SCORE(J:J) .EQ. ';') THEN
                            LCOL = J - 1
                            GO TO 51
                          ENDIF
50                      CONTINUE
51                      FORM(3:4) = NUMB26(LCOL-FCOL+1)(1:2)
                        IF ((LCOL-FCOL) .LT. 6) FORM(6:6) = FORM(4:4)

C                       ----------------Found the value for the transition
C                                       line.

                        READ (SCORE(FCOL:LCOL),FORM) LINE(L1,L2)
                        FCOL = 0
                        FORM = '(E15.7)'
                        DO 60 J=(LCOL+1),72
                          IF (SCORE(J:J) .NE. ' ' .AND.
     +                        SCORE(J:J) .NE. ';') THEN
                            FCOL = J
                            GO TO 61
                          ENDIF
60                      CONTINUE
61                      IF (FCOL .GT. 0) THEN
                          DO 70 J=FCOL,72
                            IF (SCORE(J:J) .EQ. ' ') THEN
                              LCOL = J - 1
                              GO TO 71
                            ENDIF
70                        CONTINUE
71                        FORM(3:4) = NUMB26(LCOL-FCOL+1)(1:2)
                          IF ((LCOL-FCOL) .LT. 6) FORM(6:6) = FORM(4:4)
```

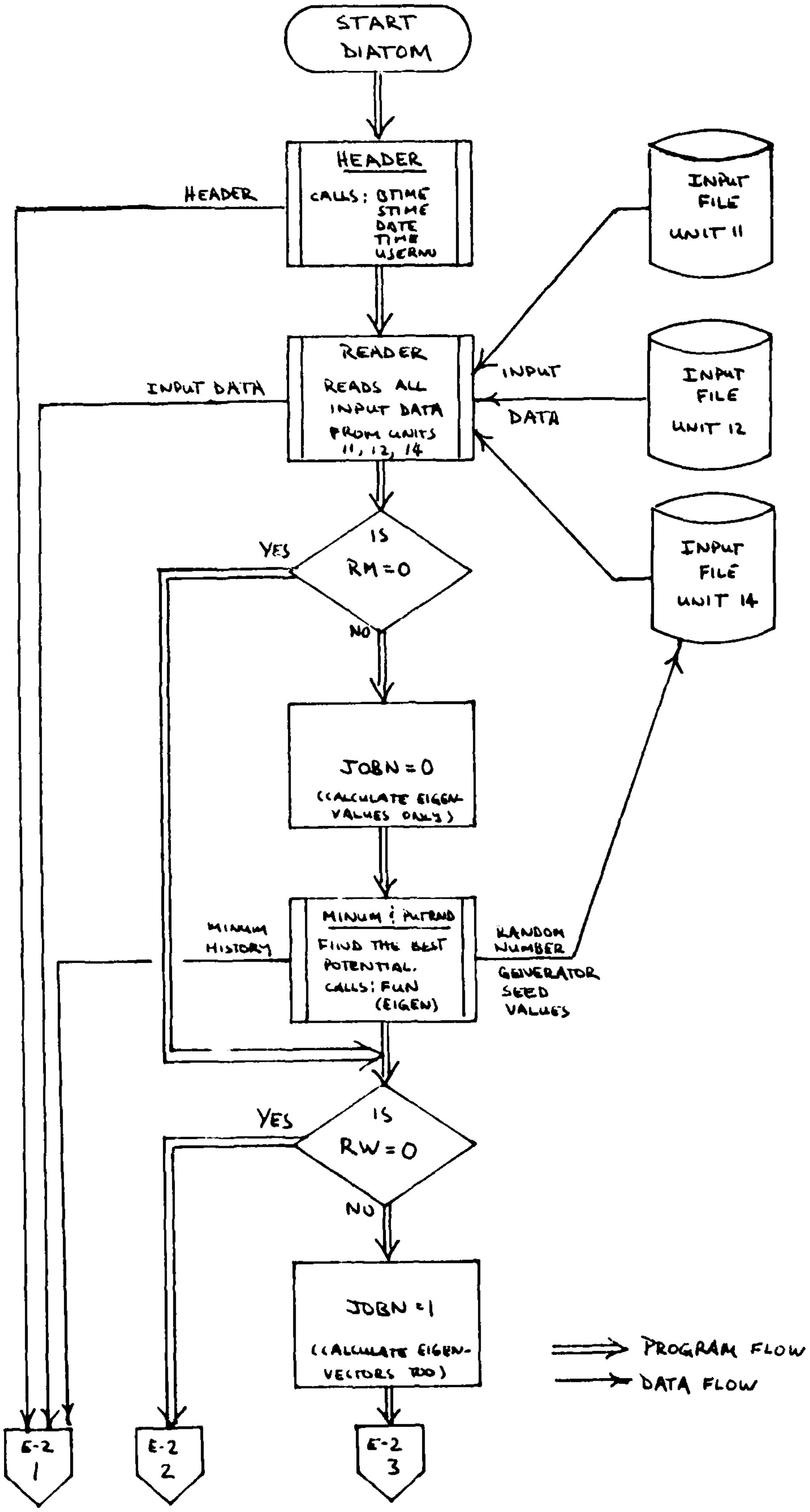
```
C                         --------------Found the value for the weighting
C                                       factor.

                          READ (SCORE(FCOL:LCOL),FORM) WEIGHT(L1,L2)
                        ELSE

C                         --------------Did not find a value for the
C                                       weighting factor, so it defaults
C                                       to 1.0

                          WEIGHT(L1,L2) = 1.0
                        ENDIF
                      ENDIF
                    ENDIF
                  ENDIF
                ENDIF
              ENDIF
            ENDIF
20    CONTINUE

C     ------------------------------Close the input file.

21    CLOSE (UNIT=11)

9999  RETURN
      END
```

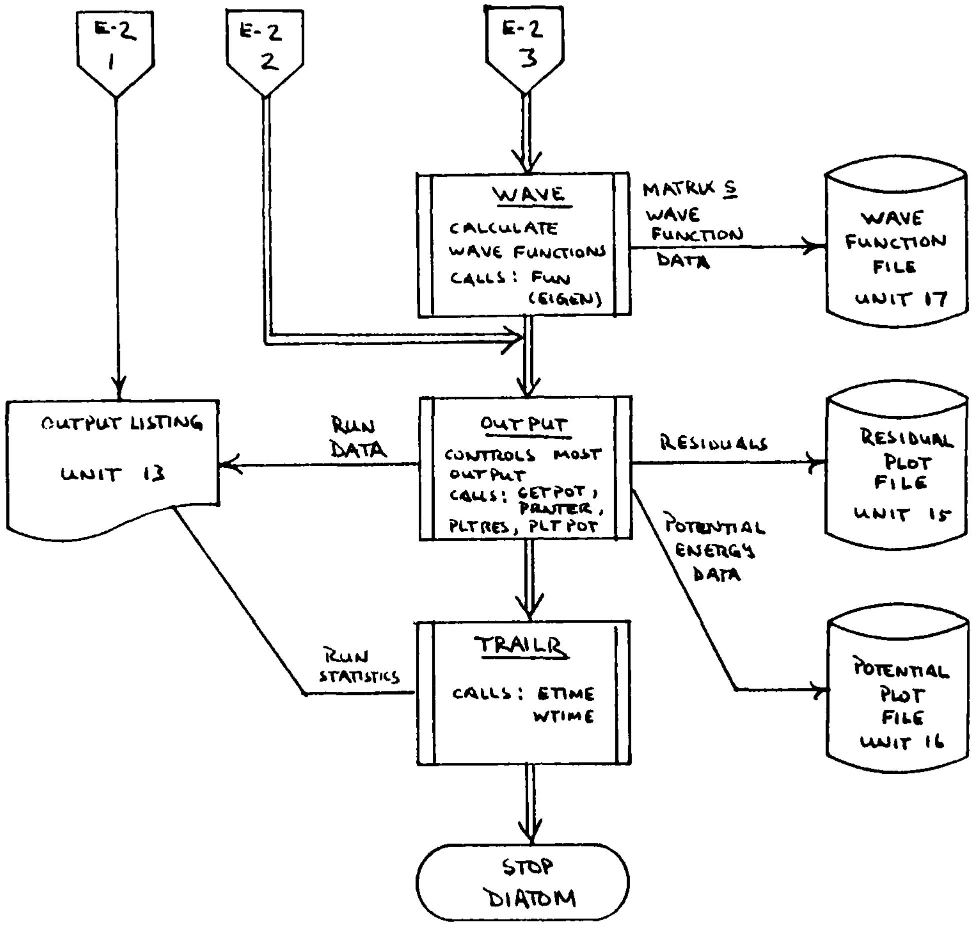
```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE LVLEFT
C
C     Version:  84.08.13
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine returns an absolute energy level number
C                   used by EFIT from the energy level number CRULVL
C                   that is relative to the electronic state CURST.
C
C-----------------------------------------------------------------------

      SUBROUTINE LVLEFT (LVLNO)

      CHARACTER*1 CURST, STLBL(10)

      INTEGER STATE(10,25), CURLVL, I, J, LVLNO, NINST(10),
     +        OFFSET, SETCNT

      COMMON /LABELS/ STLBL, CURST
      COMMON /LEVELS/ NINST, STATE, CURLVL, SETCNT

      OFFSET = 0

C     -------------------------------Match the current state with a state
C                                    in STLBL.

      DO 20 I=1,10
        IF (STLBL(I) .EQ. CURST) THEN

C         ---------------------------Now match the current level with a
C                                    level in STATE(state,level).

          DO 10 J=1,NINST(I)
            IF (STATE(I,J) .EQ. CURLVL) THEN

C             -----------------------LVLNO is the absolute level number.
C                                    The level number in CURLVL is
C                                    relative to the state in CURST.

              LVLNO = OFFSET + J
              GO TO 9999
            ENDIF
10        CONTINUE
        ELSE
          OFFSET = OFFSET + NINST(I)
        ENDIF
20    CONTINUE

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE TRLEFT
C
C     Version:  84.08.13
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  TRLEFT closes the output listing file (logical
C                   units 13 and 6) and stops the CPU and wall time
C                   use statistics.
C
C-----------------------------------------------------------------------

      SUBROUTINE TRLEFT

C     ------------------------------Shut down the run statistics.

      CALL ETIME
      CALL WTIME

C     ------------------------------Close the listing outputs.

      CLOSE (UNIT=13,STATUS='KEEP')
      CLOSE (UNIT=6)

9999  RETURN
      END
```

Appendix E

Program DIATOM Flow




E-2 1
E-2 2
E-2 3
WAVE
CALCULATE WAVE FUNCTIONS
CALLS: FUN (EIGEN)
MATRIX S WAVE FUNCTION DATA
WAVE FUNCTION FILE UNIT 17
OUTPUT LISTING
UNIT 13
RUN DATA
OUTPUT
CONTROLS MOST OUTPUT
CALLS: GETPOT, PRNTER, PLTRES, PLTPOT
RESIDUALS
RESIDUAL PLOT FILE UNIT 15
POTENTIAL ENERGY DATA
POTENTIAL PLOT FILE UNIT 16
RUN STATISTICS
TRAILR
CALLS: ETIME WTIME
STOP DIATOM

Appendix F

Program DIATOM

```
C-------------------------------------------------------------------------
C
C      Program:  DIATOM
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This program uses a finite element method to solve
C                    the Schrodinger wave equation in one dimension for
C                    diatomic molecules.  A non-linear minimization
C                    routine is used to find a set of parameters that
C                    describe the potential energy curve that best fits
C                    experimentally observed energy levels.
C
C                    I/O logical unit  6 -- output listing file
C                                     11 -- input file
C                                     12 -- energy input file
C                                     13 -- output listing file
C                                           (same as unit 6)
C                                     14 -- random number input file
C                                     15 -- residual plot file
C                                     16 -- potential plot file
C                                     17 -- wave function output file
C
C-------------------------------------------------------------------------

      PROGRAM MAIN

      CHARACTER*72 STLBL

      INTEGER   ISTP, IPRINT, JR, JG, JA, JJ, EVMCNT, PARMNO, NELMT,
     +          NOEVAL, JOBN, N, PR, PP, EVMLVL(25), NOPNTS, LEVEL,
     +          RM, RW
      INTEGER*6 IU, IX

      REAL LVM(25), STPSZE, PARM(10), PARMLO(10), PARMHI(10), A(70),
     +     BEGIN, EVAL(202), EVEC(202,202), H(20503), S(202,4),
     +     EVMW(25), RESID(25), CONST(10), NODE(101),
     +     PSIVAL(202), HBAR, MU, MIN, L(202,4)

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                 NELMT, BEGIN, HBAR, MU
      COMMON /PARMS/   PARM, PARMLO, PARMHI, CONST, PARMNO
      COMMON /EIGENS/  EVAL, EVEC, NOEVAL, JOBN
      COMMON /MATRIX/  H, S, L, N
      COMMON /WAVFUN/  NODE, PSIVAL, NOPNTS, LEVEL
      COMMON /CHRLBL/  STLBL

      EXTERNAL FUN
```

```
C     ------------------------------Open the output listing file, print
C                                   the header, and start run stats.

      CALL HEADER

C     ------------------------------Read the input files for control
C                                   parameters, input data, and random
C                                   number generator seeds.

      CALL READER (IU, IX, ISTP, IPRINT, JR, JG, JA, JJ, PR, PP, RM, RW)

      IF (RM .NE. 0) THEN

C       ----------------------------Find the set of potential energy
C                                   parameters that gives the best
C                                   least squares fit of measured
C                                   and calculated energy values.

        JOBN = 0

        CALL MINUM (PARMNO,PARM,A,FUN,IU,IX,ISTP,IPRINT,JR,JG,JA,JJ)

C       ----------------------------Put the current random number
C                                   generator seeds into a file for use
C                                   on the next execution of this
C                                   program.

        CALL PUTRND (IU, IX)

      ENDIF

      IF (RW .NE. 0) THEN

C       ----------------------------Find the normalized wave functions
C                                   that correspond with the calculated
C                                   energy levels.

        JOBN = 1

        CALL WAVE

      ELSE

        MIN = FUN(PARM)

      ENDIF

C     ------------------------------Create the output listing and files
C                                   with residual and potential plot
C                                   data, and the wave functions.

      CALL OUTPUT (PR, PP)
```

```
C       -----------------------------Close the output file and run
C                                    statistics.

      CALL TRAILER

9999  END
$ADD,HEADER
$ADD,READER
$ADD,PUTRND
$ADD,FUN
$ADD,POTENT
$ADD,EIGEN
$ADD,CETL
$ADD,FOLDZ
$ADD,FOLDX
$ADD,WAVE
$ADD,UNFOLD
$ADD,NORMAL
$ADD,OUTPUT
$ADD,GETPOT
$ADD,PRNTER
$ADD,PLTRES
$ADD,PLTPOT
$ADD,PUTWAV
$ADD,TRAILER
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE HEADER
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  HEADER opens the output listing file (logical units
C                   13 and 6) and starts CPU and wall time use
C                   statistics.
C
C-----------------------------------------------------------------------

      SUBROUTINE HEADER

      CHARACTER*8 VERSN
      CHARACTER*13 PCN

      INTEGER*3 IDATE(3), ITIME(3), IUSER(4)

      VERSN = '84.11.30'
      PCN   = 'GEP/84D-6/1.3'

C     ------------------------------Initiate the run statistics.

      CALL ETIME
      CALL STIME

C     ------------------------------Get the current date, time, and
C                                   user name for output on the header

      CALL DATE(IDATE)

      CALL TIME(ITIME)

      CALL USERNO(IUSER)

C     ------------------------------Open the output listing file and
C                                   write out the header

      OPEN (UNIT=13)
      OPEN (UNIT=6)

      WRITE (13,1001) IUSER, VERSN, IDATE, PCN, ITIME
1001  FORMAT ('1 User:  ',4A3,T51,'Air Force Institute of Technology',
     +         T110,'Version:  ',A8,
     +       /,'  Date: ',3A3,T114,'PCN:  ',A13,
     +       /,'  Time: ',3A3,T64,'DIATOM',/,/)

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C      Program:  READER
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  READER opens 3 input files (unit 11,12,13) and reads
C                    all records within.  Each record read is written to
C                    the output listing (unit 13).  Data records are
C                    marked by a '>' in column 1.  These records contain
C                    data referenced by a single key word.  All other
C                    records are considered comments.  This routine uses
C                    an internal read, eg READ (SCORE(5:19),'(E15.7)') X
C                    reads from columns 5 to 19 of the character
C                    variable SCORE using the edit descriptor E15.7 into
C                    the real variable X.
C
C-----------------------------------------------------------------------

      SUBROUTINE READER (IU,IX,ISTP,IPRINT,JR,JG,JA,JJ,PR,PP,RM,RW)

      CHARACTER*1  DIGIT(0:9)
      CHARACTER*2  LOCK(9), RLOCK, NUMBER(0:30)
      CHARACTER*9  FORM
      CHARACTER*72 SCORE, STLBL

      INTEGER EOOR(9), EVMCNT, I, IPRINT, ISTP, J, JA, PR, PP,
     +        JG, JJ, JR, NELMT, PARMNO, FCOUNT, FCOL, LCOL,
     +        EVMLVL(25), RM, RW

      INTEGER*5 IU,IX

      LOGICAL SORTED

      REAL EVM(25), PARM(10), PARMLO(10), PARMHI(10), STPSZE, BEGIN,
     +     ENDS, TEMP, RESID(25), EVMW(25), CONST(10), HBAR, MU

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                 NELMT, BEGIN, HBAR, MU
      COMMON /PARMS/  PARM, PARMLO, PARMHI, CONST, PARMNO
      COMMON /CHRLBL/ STLBL

      DATA DIGIT /'0','1','2','3','4','5','6','7','8','9'/

      DATA NUMBER /'00','01','02','03','04','05','06','07','08','09',
     +       '10','11','12','13','14','15','16','17','18','19','20',
     +            '21','22','23','24','25','26','27','28','29','30'/

      DATA LOCK /'NE','IS','IP','JR','JG','JA','JJ','PR','PP'/
```

```
C       ---------------------------------Initialize some variables.

      DO 10 I=1,9
        ICOR(I) = 0
10    CONTINUE

      PARMNO = 0

C       ---------------------------------Open the Control Parameter Input
C                                        File -- logical unit 11.

      OPEN (UNIT=11)

      WRITE (13,1001)
1001  FORMAT ('0',T24,'Control Parameter Input File',/,
     +        ' ',T24,'------- --------- ----- ----')

      DO 40 I=1,1000

C         -----------------------------Transfer a record from the input
C                                      file to the buffer SCORE.

        READ (11,1101,END=41) SCORE
1101    FORMAT (A72)

        WRITE (13,1102) SCORE
1102    FORMAT (' ',A72)

C         -----------------------------If SCORE(1:1) is a '>', then SCORE
C                                      should contain data and a valid
C                                      keyword.  So, see if SCORE(2:3)
C                                      does contain a valid keyword.  If it
C                                      does, transfer the data from SCORE
C                                      to the input variable.

        IF (SCORE(1:1) .EQ. '>') THEN
          IF (SCORE(2:3) .EQ. 'HB') THEN
            READ (SCORE(5:19),'(E15.7)') HBAR
          ELSE
            IF (SCORE(2:3) .EQ. 'MU') THEN
              READ (SCORE(5:19),'(E15.7)') MU
            ELSE
              IF (SCORE(2:3) .EQ. 'BG') THEN
                READ (SCORE(5:19),'(E15.7)') BEGIN
              ELSE
                IF (SCORE(2:3) .EQ. 'EN') THEN
                  READ (SCORE(5:19),'(E15.7)') ENDS
                ELSE
                  IF (SCORE(2:3) .EQ. 'RM') THEN
                    READ (SCORE(5:19),'(I15)') RM
                  ELSE
                    IF (SCORE(2:3) .EQ. 'RW') THEN
                      READ (SCORE(5:19),'(I15)') RW
                    ELSE
```

```
              IF (SCORE(2:2) .EQ. 'P') THEN
                DO 20 J=1,10
                  RLOCK(1:1) = 'P'
                  RLOCK(2:2) = NUMBER(J)(2:2)
                  IF (SCORE(2:3) .EQ. RLOCK) THEN
                    READ (SCORE(5:19),'(E15.7)') PARM(J)
                    PARMNO = PARMNO + 1
                    GO TO 40
                  ENDIF
20              CONTINUE
              ELSE
                IF (SCORE(2:2) .EQ. 'U') THEN
                  DO 21 J=1,10
                    RLOCK(1:1) = 'U'
                    RLOCK(2:2) = NUMBER(J)(2:2)
                    IF (SCORE(2:3) .EQ. RLOCK) THEN
                      READ (SCORE(5:19),'(E15.7)') PARMHI(J)
                      GO TO 40
                    ENDIF
21                CONTINUE
                ELSE
                  IF (SCORE(2:2) .EQ. 'L') THEN
                    DO 22 J=1,10
                      RLOCK(1:1) = 'L'
                      RLOCK(2:2) = NUMBER(J)(2:2)
                      IF (SCORE(2:3) .EQ. RLOCK) THEN
                        READ (SCORE(5:19),'(E15.7)') PARMLO(J)
                        GO TO 40
                      ENDIF
22                  CONTINUE
                  ELSE
                    IF (SCORE(2:2) .EQ. 'C') THEN
                      DO 23 J=1,10
                        RLOCK(1:1) = 'C'
                        RLOCK(2:2) = NUMBER(J)(2:2)
                        IF (SCORE(2:3) .EQ. RLOCK) THEN
                          READ (SCORE(5:19),'(E15.7)') CONST(J)
                          GO TO 40
                        ENDIF
23                    CONTINUE
                    ELSE
                      DO 30 K=1,9
                        IF (SCORE(2:3) .EQ. LOCK(K)) THEN
                          READ (SCORE(5:19),'(I15)') DOOR(K)
                        ENDIF
30                    CONTINUE
                    ENDIF
                  ENDIF
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
```

```
C----------------------------------------------------------------------
C
C      Program:  SUBROUTINE UNFOLD
C
C      Version:  03.07.84
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  UNFOLD recovers the eigenvectors v of Hv-eSv=0
C                    from the eigenvectors y of Xy-ey=0 where e are
C                    eigenvalues.
C
C----------------------------------------------------------------------

      SUBROUTINE UNFOLD

      INTEGER I, J, K, N, NOEVAL

      REAL    EVAL(202), EVEC(202,202), H(20503),
     +        S(202,4), SUM, L(202,4)

      COMMON /EIGENS/ EVAL, EVEC, NOEVAL, JOEN
      COMMON /MATRIX/ H, S, L, N

      DO 30 I=N,1,-1
        DO 20 J=1,NOEVAL
          SUM = 0
          DO 10 K=1,3
            IF ((I+K) .LE. N) THEN
              SUM = SUM + (L((I+K),(4-K)) * EVEC((I+K),J))
            ENDIF
10        CONTINUE
          EVEC(I,J) = (EVEC(I,J) - SUM) / L(I,4)
20      CONTINUE
30    CONTINUE

9999  RETURN
      END
```

```
C----------------------------------------------------------------------
C
C      Program:  SUBROUTINE FOLDX
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine finds a matrix X where Z = XL(transpose)
C                    which is stored in place of Z in the array H.
C
C----------------------------------------------------------------------

      SUBROUTINE FOLDX

      INTEGER I, J, K, KK, LL, LLL, N

      REAL    H(20503), L(202,4), S(202,4), SUM

      COMMON /MATRIX/ H, S, L, N

      DO 30 J=1,N
        DO 20 I=J,N
          SUM = 0
          DO 10 K=1,3
            LL  = J - K
            LLL = 4 - K
            IF (LL .GT. 0) THEN
              KK  = (I * (I-1) / 2) + LL
              SUM = SUM + (H(KK) * L(J,LLL))
            ENDIF
10        CONTINUE
          K    = (I * (I-1) / 2) + J
          H(K) = (H(K) - SUM) / L(J,4)
20      CONTINUE
30    CONTINUE

9999  RETURN
      END
```

```
C ---------------------------------------------------------------------
C
C      Program:  SUBROUTINE FOLDZ
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine finds a matrix Z such that H = LZ.
C                    Matrix Z is stored in place of H in the array H.
C
C---------------------------------------------------------------------

      SUBROUTINE FOLDZ

      INTEGER CNT, I, II, J, K, KK, LL, N

      REAL    H(20503), H3B(202,3), L(202,4), S(202,4), SUM

      COMMON /MATRIX/ H, S, L, N

      DO 30 I=1,N
        IF ((I+3) .LT. N) THEN
          CNT = I + 3
        ELSE
          CNT = N
        ENDIF
        DO 20 J=1,CNT
          SUM = 0
          DO 10 K=1,3
            LL = 4 - K
            II = I - K
            IF (II .GT. 0) THEN
              IF (II .GE. J) THEN
                KK  = (II * (II-1) / 2) + J
                SUM = SUM + (L(I,LL) * H(KK))
              ELSE
                IF ((J-II) .LE. 3) THEN
                  SUM = SUM + (L(I,LL) * H3B(II,(J-II)))
                ENDIF
              ENDIF
            ENDIF
10        CONTINUE
          IF (I .GE. J) THEN
            K    = (I * (I-1) / 2) + J
            H(K) = (H(K) - SUM) / L(I,4)
          ELSE
            K            = (J * (J-1) / 2) + 1
            H3B(I,(J-I)) = (H(K) - SUM) / L(I,4)
          ENDIF
20      CONTINUE
30    CONTINUE

9999  RETURN
      END
```

```
C----------------------------------------------------------------------
C
C      Program:  SUBROUTINE GETL
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine decomposes the matrix S into a matrix L
C                    such that S = LL(transpose).
C
C----------------------------------------------------------------------

      SUBROUTINE GETL

      INTEGER I, II, ILMT, J, LL, N

      REAL H(20503), S(202,4), SUM, L(202,4)

      COMMON /MATRIX/ H, S, L, N

      ILMT = 4

      DO 20 J=1,N
        DO 10 I=J,ILMT
          SUM = 0
          IF (I .EQ. J) THEN
            SUM    = L(I,1)**2 + L(I,2)**2 + L(I,3)**2
            L(I,4) = SQRT(S(I,4) - SUM)
          ELSE
            LL = 4 - I + J
            IF (LL .EQ. 2) THEN
              SUM = L(I,1) * L(J,3)
            ELSE
              IF (LL .EQ. 3) THEN
                SUM = L(I,1)*L(J,2) + L(I,2)*L(J,3)
              ENDIF
            ENDIF
            L(I,LL) = (S(I,LL) - SUM) / L(J,4)
          ENDIF
10      CONTINUE

        II = MOD(J,2)
        IF (II .EQ. 0) THEN
          ILMT = J + 4
        ELSE
          ILMT = J + 3
        ENDIF
20    CONTINUE

9999  RETURN
      END
```

```
      CALL FOLDX

C     -------------------------------Tridiagonalize matrix X (stored in
C                                    H) as matrix T (stored in EVAL, E)

      CALL EHOUSS (H, N, EVAL, E, E2)

C     -------------------------------Find eigenvalues and optionally
C                                    (JOBN = 1) the eigenvectors of the
C                                    problem:    Xy - ey = 0

      NOEVAL = 25
      IF (NOEVAL .GT. N) NOEVAL = N

      IF (JOBN .EQ. 0) THEN
        ISW = 0

C       -----------------------------Find the lowest NOEVAL eigenvalues
C                                    of matrix T.

        CALL EQRT1S (EVAL, E2, N, NOEVAL, ISW, IER)

      ELSE
        IZ = 202
        M1 = 1

        DO 10 I=1,N
          DO 5 J=1,N
            EVEC(I,J) = 0.0
5         CONTINUE
          EVEC(I,I) = 1.0
10      CONTINUE

C       ---------------- -----------Find eigenvalues and eigenvectors
C                                    matrix T.

        CALL EQRT2S (EVAL, E, N, EVEC, IZ, IER)

C       -----------------------------Recover the eigenvectors of matrix X
C                                    from those of matrix T.

        CALL EHOBKS (H, N, M1, NOEVAL, EVEC, IZ)

C       -----------------------------UNFOLD recovers the eigenvectors v
C                                    of:       Hv - eSv = 0
C                                    from the eigenvectors y computed in
C                                    VECTOR of:       Xy - ey = 0
C                                    where e = the eigenvalues.

        CALL UNFOLD

      ENDIF

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE EIGEN
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine finds the eigenvalues and optionally the
C                   eigenvectors (JOBN=1) of the generalized eigenvalue
C                   problem Hv-eSv=0.  This is done using routines GETL,
C                   FOLDZ, FOLDX, UNFOLD, and IMSL routines EHOUSS,
C                   EQRT1S, EQRT2S, EHOBKS.
C
C-----------------------------------------------------------------------

      SUBROUTINE EIGEN

      INTEGER I, IER, ISW, IZ, JOBN, N, NOEVAL, M, M1

      LOGICAL FIRST

      REAL    EVAL(202), EVEC(202,202), H(20503),
     +        S(202,4), L(202,4), SUM, E(202), E2(202)

      COMMON /EIGENS/ EVAL, EVEC, NOEVAL, JOBN
      COMMON /MATRIX/ H, S, L, N

      DATA FIRST /.TRUE./

C     ------------------------------Use the Cholesky decomposition to
C                                   get a matrix L such
C                                   that:    S = LL(transpose)
C                                   This only has to be done once since
C                                   S doesn't change.

      IF (FIRST) THEN
        CALL GETL
        FIRST = .FALSE.
      ENDIF

C     ------------------------------Get Z such that:   H = LZ

      CALL FOLDZ

C     ------------------------------Get X such that:  Z = XL(transpose)
C                                   the problem:   Hv - eSv = 0
C                                   now becomes:   Xy - ey  = 0
C                                   where:  e = eigenvalues
C                                           v = eigenvectors
C                                           y = eigenvectors such that:
C                                                 y = L(transpose)v
```

```
      DO 110 I=1,ELMT
        IF (EVM(I) .EQ. 0.0) THEN
          RESID(I) = 0
        ELSE
          RESID(I) = EVM(I) - EVAL(I)
          FUN      = FUN + (EVMW(I) * RESID(I)**2) * 0.5
        ENDIF
110   CONTINUE

      IF (FUN .GT. 1E10) FUN = 1E10

9999  RETURN
      END
```

```
      JJLESS = 0
      DO 30 J=1,10
C           ------------------------------Use the potential values and matrix
C                                         seeds to build the sub-matricies.

        V1(J) =  P0 * V1H(J)
        V2(J) = (3*P0 + DP0*STPSZE) * V2H(J)
        V3(J) =  P1 * V3H(J)
        V4(J) = (3*P1 - DP1*STPSZE) * V4H(J)

        KK = (II * (II-1) / 2) + JJ
        LL = 4 - II + JJ

C           ------------------------------Load the sub-matrix values into H
C                                         and S.

        H(KK) = H(KK) + T(J) + V1(J) + V2(J) + V3(J) + V4(J)

        IF (FIRST) THEN
          S(II,LL) = S(II,LL) + SH(J)
        ENDIF

        IF (II .EQ. JJ) THEN
          II     = II + 1
          JJ     = JJ - JJLESS
          JJLESS = JJLESS + 1
        ELSE
          JJ = JJ + 1
        ENDIF
30    CONTINUE
      II = II - 2
      JJ = JJ + 2
40    CONTINUE

      FIRST = .FALSE.

C     ------------------------------------Subroutine EIGEN solves the general
C                                         eigenvalue problem:
C                                                   Hv - eSv = 0
C                                         where:  H is the H (energy) matrix
C                                                 v is the eigenvector
C                                                 e is the eigenvalue
C                                                 S is the normalizing matrix


      CALL EIGEN

      FUN  = 0.0
      ELMT = EVMCNT
      IF (NOEVAL .LT. EVMCNT) ELMT = NOEVAL

C     ------------------------------------Compute the residuals and the value
C                                         of FUN.
```

```
        VHH(3)  =    2880 * STPSZE**3 / 3628800
        VHH(4)  =   50400 * STPSZE    / 3628800
        VHH(5)  =   12960 * STPSZE**2 / 3628800
        VHH(6)  =  139680 * STPSZE    / 3628800
        VHH(7)  =  -11520 * STPSZE**2 / 3628800
        VHH(8)  =   -2880 * STPSZE**3 / 3628800
        VHH(9)  =  -20040 * STPSZE**2 / 3628800
        VHH(10) =    4320 * STPSZE**3 / 3628800

C       ----------------------------N is the number of members on the
C                                   main diagonals of H and S.

        N = (2 * NELMT) + 2
      ENDIF

C       ----------------------------Clear out the arrays that H and S
C                                   are stored in.  Do this only the
C                                   first time for S.

      DO 20 J=1,N
        DO 10 I=J,N
          KK    = (I * (I-1) / 2) + J
          LL    = 4 - I + J
          H(KK) = 0
          IF (FIRST .AND. LL .GT. 0) THEN
            S(I,LL)    = 0
          ENDIF
10      CONTINUE
20    CONTINUE

C     ------------------------------Build the sub-matricies that are
C                                   pieced into matricies H and S
C                                   from the seeds.  This is done.
C                                   in an iterative manner, once for
C                                   each step along the grid.

      FSTLP = .TRUE.

      DO 40 I=1,NELMT
        R0 = BEGIN + (STPSZE * (I-1))
        R1 = BEGIN + (STPSZE * I)

C       ----------------------------Subroutine POTENT returns the
C                                   value of, and slope of the model
C                                   potential function at each end of
C                                   the current grid element (R0,R1).

        CALL POTENT (R0, R1, P0, DP0, P1, DP1)

        IF (FSTLP) THEN
          II    = 1
          JJ    = 1
          FSTLP = .FALSE.
        ENDIF
```

```
      SH(9)  = -264 * STPSZE**2 / 5040
      SH(10) =   48 * STPSZE**3 / 5040

      T(1)  = 144 / (240 * STPSZE)
      T(2)  = 12.0 / 240
      T(3)  = (16 * STPSZE) / 240
      T(4)  = -144.0 / (240 * STPSZE)
      T(5)  = -12.0 / 240
      T(6)  = T(1)
      T(7)  = T(2)
      T(8)  = (-4 * STPSZE) / 240
      T(9)  = T(5)
      T(10) = T(3)

      PRE = (HBAR*HBAR)/MU
      DO 5 I=1,10
        T(I) = PRE * T(I)
5     CONTINUE

      V1H(1)  = 695520 * STPSZE    / 3628800
      V1H(2)  =  70560 * STPSZE**2 / 3628800
      V1H(3)  =  10080 * STPSZE**3 / 3628800
      V1H(4)  =  82080 * STPSZE    / 3628800
      V1H(5)  =  15840 * STPSZE**2 / 3628800
      V1H(6)  =  47520 * STPSZE    / 3628800
      V1H(7)  = -23040 * STPSZE**2 / 3628800
      V1H(8)  =  -4320 * STPSZE**3 / 3628800
      V1H(9)  = -11520 * STPSZE**2 / 3628800
      V1H(10) =   2880 * STPSZE**3 / 3628800

      V2H(1)  = 139680 * STPSZE    / 3628800
      V2H(2)  =  23040 * STPSZE**2 / 3628800
      V2H(3)  =   4320 * STPSZE**3 / 3628800
      V2H(4)  =  50400 * STPSZE    / 3628800
      V2H(5)  =  11520 * STPSZE**2 / 3628800
      V2H(6)  =  61920 * STPSZE    / 3628800
      V2H(7)  = -12960 * STPSZE**2 / 3628800
      V2H(8)  =  -2880 * STPSZE**3 / 3628800
      V2H(9)  = -12960 * STPSZE**2 / 3628800
      V2H(10) =   2880 * STPSZE**3 / 3628800

      V3H(1)  =  47520 * STPSZE    / 3628800
      V3H(2)  =  11520 * STPSZE**2 / 3628800
      V3H(3)  =   2880 * STPSZE**3 / 3628800
      V3H(4)  =  82080 * STPSZE    / 3628800
      V3H(5)  =  23040 * STPSZE**2 / 3628800
      V3H(6)  = 695520 * STPSZE    / 3628800
      V3H(7)  = -15840 * STPSZE**2 / 3628800
      V3H(8)  =  -4320 * STPSZE**3 / 3628800
      V3H(9)  = -70560 * STPSZE**2 / 3628800
      V3H(10) =  10080 * STPSZE**3 / 3628800

      V4H(1)  =  61920 * STPSZE    / 3628800
      V4H(2)  =  12960 * STPSZE**2 / 3628800
```

```
C-----------------------------------------------------------------------
C
C      Program:  FUNCTION FUN
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine solves the Schrodinger wave equation
C                    using a set of potential energy parameters to build
C                    the wave equation in matrix form.  Then EIGEN is
C                    used to solve the eigenvalue problem for the eigen-
C                    values.  These eigenvalues are compared in a weight-
C                    ed least squares sense with the observed energy
C                    levels.  The value of FUN returned is the sum of the
C                    weighted residuals squared.
C
C-----------------------------------------------------------------------

       REAL FUNCTION FUN (PARM)

       INTEGER EVMCNT, I, IER,  II, J, JJ, JJLESS, JOBN, KK, LL, N,
     +         NOEVAL, NELMT, ELMT, EVMLVL(25)

       LOGICAL FIRST, FSTLP

       REAL EVM(25), T(10), V1H(10), V2H(10), V3H(10), V4H(10), V1(10),
     +      V2(10), V3(10), V4(10), STPSZE, R0, R1, H(20503), S(202,4),
     +      SH(10), P0, DP0, P1, DP1, PARM(10), EVAL(202), EVEC(202,202),
     +      DELTA, RESID(25), BEGIN, EVMW(25), HBAR, MU, PRE, L(202,4)

       COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                  NELMT, BEGIN, HBAR, MU
       COMMON /EIGENS/ EVAL, EVEC, NOEVAL, JOBN
       COMMON /MATRIX/ H, S, L, N

       DATA FIRST /.TRUE./

C      ------------------------------Load the arrays used to build the
C                                    matricies H, S, and NORM.  These
C                                    will not change, do this once only.

       IF (FIRST) THEN
         SH(1)  = 1872 * STPSZE    / 5040
         SH(2)  =  264 * STPSZE**2 / 5040
         SH(3)  =   48 * STPSZE**3 / 5040
         SH(4)  =  648 * STPSZE    / 5040
         SH(5)  =  156 * STPSZE**2 / 5040
         SH(6)  = 1872 * STPSZE    / 5040
         SH(7)  = -156 * STPSZE**2 / 5040
         SH(8)  =  -36 * STPSZE**3 / 5040
```

```
                  ENDIF
                ENDIF
210           CONTINUE
              ENDIF
            ENDIF
          ENDIF
100   CONTINUE

C     ------------------------------Close the input file.

101   CLOSE (UNIT=12)

C     ------------------------------Open the Random Number Generator
C                                   Seed file -- logical unit 14.

      OPEN (UNIT=14)

      WRITE (13,1304)
1304  FORMAT ('0',T21,'Random Number Generator Seed File',/,
     +        ' ',T21,'------ ------ --------- ---- ----')

      DO 120 I=1,1000

C       ----------------------------Transfer a record from the input
C                                   file to the buffer SCORE.

        READ (14,1401,END=121) SCORE
1401    FORMAT (A72)

        WRITE (13,1302) SCORE

C       ----------------------------If SCORE(1:1) is a '>', then SCORE
C                                   should contain data and a valid
C                                   keyword.  So, see if SCORE(2:3)
C                                   does contain a valid keyword.  If it
C                                   does, transfer the data from SCORE
C                                   to the input variable.

        IF (SCORE(1:1) .EQ. '>') THEN
          IF (SCORE(2:3) .EQ. 'IU') THEN
            READ (SCORE(5:19),'(I15)') IU
          ELSE
            IF (SCORE(2:3) .EQ. 'IX') THEN
              READ (SCORE(5:19),'(I15)') IX
            ENDIF
          ENDIF
        ENDIF
120   CONTINUE

C     ------------------------------Close the input file.

121   CLOSE (UNIT=14)

9999  RETURN
      END
```

```
C                                      does contain a valid keyword.  If it
C                                      does, transfer the data from SCORE
C                                      to the input variable.

          IF (SCORE(1:1) .EQ. '>') THEN
            IF (SCORE(2:4) .EQ. 'LBL') THEN
              STLBL(1:72) = SCORE(6:77)
            ELSE
              IF (SCORE(2:2) .EQ. 'V')  THEN
                DO 210 K=0,25
                  IF (SCORE(3:4) .EQ. NUMBER(K)) THEN
                    EVMCNT = EVMCNT + 1
                    EVMLVL(EVMCNT) = K
                    FCOL = 0
                    LCOL = 0
                    FORM = '(E15.7)'
                    DO 60 J=6,72
                      IF (SCORE(J:J) .NE. ' ' .AND.
     +                    SCORE(J:J) .NE. ';') THEN
                        FCOL = J
                        GO TO 61
                      ENDIF
60                  CONTINUE
61                  IF (FCOL .GT. 0) THEN
                      DO 70 J=FCOL,72
                        IF (SCORE(J:J) .EQ. ' ' .OR.
     +                      SCORE(J:J) .EQ. ';') THEN
                          LCOL = J - 1
                          GO TO 71
                        ENDIF
70                    CONTINUE
71                    FORM(3:4) = NUMBER(LCOL-FCOL+1)(1:2)
                      IF ((LCOL-FCOL) .LT. 6) FORM(6:6) = FORM(4:4)
                      READ (SCORE(FCOL:LCOL),FORM) EVM(EVMCNT)
                      FCOL = 0
                      FORM = '(E15.7)'
                      DO 80 J=(LCOL+1),72
                        IF (SCORE(J:J) .NE. ' ' .AND.
     +                      SCORE(J:J) .NE. ';') THEN
                          FCOL = J
                          GO TO 81
                        ENDIF
80                    CONTINUE
81                    IF (FCOL .GT. 0) THEN
                        DO 90 J=FCOL,72
                          IF (SCORE(J:J) .EQ. ' ') THEN
                            LCOL = J - 1
                            GO TO 91
                          ENDIF
90                      CONTINUE
91                      FORM(3:4) = NUMBER(LCOL-FCOL+1)(1:2)
                        IF ((LCOL-FCOL) .LT. 6) FORM(6:6) = FORM(4:4)
                        READ (SCORE(FCOL:LCOL),FORM) EVMW(EVMCNT)
                      ELSE
                        EVMW(EVMCNT) = 1.0
                      ENDIF
```

```
            ENDIF
          ENDIF
        ENDIF
40    CONTINUE

C             ----------- -------------Close the input file.

41    CLOSE (UNIT=11)

      NELMT   = DOOR(1)
      ISTP    = DOOR(2)
      IPRINT  = DOOR(3)
      JR      = DOOR(4)
      JG      = DOOR(5)
      JA      = DOOR(6)
      JJ      = DOOR(7)
      PR      = DOOR(8)
      PP      = DOOR(9)
      STPSZE = (ENDS - BEGIN) / NELMT

      DO 46 I=1,PARMNO
        PARMHI(I) = PARMHI(I) - PARMLO(I)
        PARM(I)   = (PARM(I) - PARMLO(I)) / PARMHI(I)
46    CONTINUE

      DO 50 I=1,25
        EVM(I) = 0.0
        EVMLVL(I) = 0.0
50    CONTINUE

      EVMCNT = 0

C     ------------------------------Open the Measured Energy Data File
C                                   -- logical unit 12.

      OPEN (UNIT=12)

      WRITE (13,1303)
1303  FORMAT ('0',T25,'Measured Energy Data File',/,
     +        ' ',T25,'-------- ------ ---- ----')

      DO 100 I=1,1000

C       ----------------------------Transfer a record from the input
C                                   file to the buffer SCORE.

        READ (12,1201,END=101) SCORE
1201    FORMAT (A72)

        WRITE (13,1302) SCORE

C       ----------- ----------------If SCORE(1:1) is a '>', then SCORE
C                                   should contain data and a valid
C                                   keyword.  So, see if SCORE(2:4)
```

```
C-----------------------------------------------------------------------
C
C      Program:  SUBROUTINE PUTRND
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine stores the random number generator seed
C                    values used by MINUM in a file (unit 14) for future
C                    reference.
C
C-----------------------------------------------------------------------

      SUBROUTINE PUTRND (IU, IX)

      INTEGER*8 IU, IX

C      ------------------------------Put the current random number
C                                    generator seed values into a file
C                                    for future reference.

      OPEN (UNIT=14)

      WRITE (14,1401)
      WRITE (14,1402)
      WRITE (14,1403)
      WRITE (14,1401)
      WRITE (14,1404) IU
      WRITE (14,1405) IX

1401  FORMAT (7('**********'),'**')
1402  FORMAT ('***        Seeds for a random number generator in MINUM',
     +'              ***')
1403  FORMAT ('***                        called by program DIATOM       ',
     +'              ***')
1404  FORMAT ('>IU=',I15)
1405  FORMAT ('>IX=',I15)

C      ------------------------------Write the end-of-file mark and
C                                    close the file.

      ENDFILE (UNIT=14)

      CLOSE (UNIT=14,STATUS='KEEP')

9999  RETURN
      END
```

```
C------------------------------------------------------------------------
C
C     Program:  SUBROUTINE WAVE
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  WAVE controls the execution of FUN (and therefore
C                   EIGEN), NORMAL, and PUTWAV to calcu-
C                   late and store the normalized wave functions.  This
C                   is done in an iterative fashion for each wave func-
C                   tion once the eigenvectors are returned from FUN.
C
C------------------------------------------------------------------------

      SUBROUTINE WAVE

      CHARACTER*72 STLBL

      INTEGER   ELMT, I, NOEVAL, JOBN, N, NELMT, EVMCNT, EVMLVL(25),
     +          NOPNTS, LEVEL, HUNDRD

      REAL EVM(25), STPSZE, NODE(101), PSIVAL(202),
     +     BEGIN, EVAL(202), EVEC(202,202), H(20503), S(202,4),
     +     EVMW(25), RESID(25), MIN, HBAR, MU, L(202,4)

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                NELMT, BEGIN, HBAR, MU
      COMMON /EIGENS/ EVAL, EVEC, NOEVAL, JOBN
      COMMON /MATRIX/ H, S, L, N
      COMMON /WAVFUN/ NODE, PSIVAL, NOPNTS, LEVEL
      COMMON /CHRLBL/ STLBL

C     ------------------------------Call function FUN using
C                                   MINUM's "best" potential energy
C                                   parameter set to get the
C                                   eigenvalues and vectors.

      MIN = FUN(PARM)

      WRITE (13,1301) MIN
 1301 FORMAT ('0The sum of residuals squared is:',G15.7)

      NODE(1) = BEGIN
      DO 10 I=2,(NELMT+1)
        NODE(I) = NODE(I-1) + STPSZE
 10   CONTINUE

      NOPNTS = N
      HUNDRD = 100
```

```
      ELMT    = 25
      IF (NOEVAL .LT. 25) ELMT = NOEVAL

C     ------------------------------Open the wave function output file.

      OPEN (UNIT=17)

C     ------------------------------Write the label/comment read in from
C                                   unit 12 in READER.

      WRITE (17,1701) STLBL(1:65)
1701  FORMAT ('>LABL=',A65)

C     ------------------------------Write the matrix S to the wave
C                                   function output file.

      DO 50 I=1,N
        WRITE (17,1702) I, (S(I,J),J=1,4)
1702    FORMAT ('>S',I3,4G15.7)
50    CONTINUE

C     ------------------------------Normalize each eigenvector (wave
C                                   function) stored in EVEC.  Then
C                                   write it to the wave function
C                                   output file.

      DO 100 LEVEL=1,ELMT
        DO 110 I=1,N
          PSIVAL(I) = EVEC(I,LEVEL)
110     CONTINUE

C       -- -------------------------NORMAL normalizes the eigenvectors
C                                   (wave function values).

        CALL NORMAL

C       ----------------------------PUTWAV writes the wave function to
C                                   the output file.

        CALL PUTWAV

100   CONTINUE

C     ------------------------------Write an End-Of-File mark and close
C                                   the output file.

      ENDFILE (UNIT=17)
      CLOSE (UNIT=17)

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE NORMAL
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine normalizes the wave function referenced
C                   by LEVEL.
C
C-----------------------------------------------------------------------

      SUBROUTINE NORMAL

      INTEGER I, J, K, KK, N, NOEVAL, NOPNTS, LEVEL, JOBN,
     +        EVMLVL(25), EVMCNT, NELMT, IER, I1, I3, I202

      REAL EVM(25), STPSZE, NODE(101), PSIVAL(202),
     +     BEGIN, EVAL(202), EVEC(202,202), H(20503), S(202,4),
     +     EVMW(25), RESID(25), AREA, SUM,
     +     A(202), HBAR, MU, L(202,4)

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                 NELMT, BEGIN, HBAR, MU
      COMMON /EIGENS/ EVAL, EVEC, NOEVAL, JOBN
      COMMON /MATRIX/ H, S, L, N
      COMMON /WAVFUN/ NODE, PSIVAL, NOPNTS, LEVEL

      DATA I1, I3, I202 /1, 3, 202/

      AREA = 0.0

C     ------------------------------Multiply S*PSIVAL = A
C
C                                   VMULQF is an IMSL routine for matrix
C                                   multiplication.
C                                   (Band Symmetric storage mode times
C                                   Full storage mode.)

      CALL VMULQF (S, N, I3, I202, PSIVAL, I1, I202, A, I202)

C     ------------------------------Multiply PSIVAL*A = AREA
C
C                                   VMULFF is an IMSL routine for matrix
C                                   multiplication.
C                                   (Full storage mode times Full
C                                   storage mode.)

      CALL VMULFF (PSIVAL, A, I1, N, I1, I1, I202, AREA,I1, IER)
```

```
C     ------------------------------Normalize the wave function.

      DO 50 I=1,N
        PSIVAL(I) = PSIVAL(I) / SQRT(ABS(AREA))
50    CONTINUE

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE PUTWAV
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  PUTWAV writes the value of the current wavefunction
C                   the node value, and the cubic spline coefficients
C                   to a file (logical unit 17).
C
C-----------------------------------------------------------------------

      SUBROUTINE PUTWAV

      INTEGER NOPNTS, LEVEL, VIBLVL, I, J

      REAL    NODE(101), PSIVAL(202)

      COMMON /WAVFUN/ NODE, PSIVAL, NOPNTS, LEVEL

C     ------------------------------Write the vibrational quantum number
C                                   , a count number, wave function
C                                   value to each record.

      VIBLVL = LEVEL - 1
      DO 10 I=1,NOPNTS
        WRITE (17,1701) VIBLVL, I, PSIVAL(I)
1701    FORMAT ('>',2I4,G15.7)
10    CONTINUE

9999  RETURN
      END
```

```
C-----------------------------------------------------------------
C
C     Program:  SUBROUTINE OUTPUT
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  OUTPUT controls the execution of output routines
C                   GETPOT, PRNTER, PLTRES, and PLTPOT.
C
C-----------------------------------------------------------------

      SUBROUTINE OUTPUT (PR, PP)

      INTEGER PR, PP

      REAL R(1000), POTVAL(1000)

C     ------------------------------Subroutine GETPOT calculates the
C                                   value of the potential energy model
C                                   at 1000 grid points.

      CALL GETPOT (R, POTVAL)

C     ------------------------------Subroutine PRNTER writes to the
C                                   output listing file.

      CALL PRNTER (R, POTVAL)

C     ------------------------------Use PLTRES to create a residual plot
C                                   file if the user set PR=1 in the
C                                   input file (unit 11).

      IF (PR .NE. 0) THEN
        CALL PLTRES
      ENDIF

C     ------------------------------Use PLTPOT to create a potential
C                                   energy plot file if the user set
C                                   PP=1 in the input file (unit 11).

      IF (PP .NE. 0) THEN
        CALL PLTPOT (R, POTVAL)
      ENDIF

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C      Program:  SUBROUTINE GETPOT
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine calculates the value of the potential
C                    energy model at 1000 grid points using the best
C                    parameter set.
C
C-----------------------------------------------------------------------

      SUBROUTINE GETPOT (R, POTVAL)

      INTEGER EVMCNT, I, NELMT, EVMLVL(25)

      REAL    BEGIN, STPSZE, R(1000), POTVAL(1000), RO, R1, PO, DPO,
     +        P1, DP1, EVM(25), EVMW(25), RESID(25), STEP, HBAR, MU

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                 NELMT, BEGIN, HBAR, MU

      STEP = (STPSZE * NELMT) / 1000.0

      DO 10 I=1,999
        RO = BEGIN + (STEP * (I-1))
        R1 = RO + STEP

C       ----------------------------POTENT returns the value of the
C                                   potential function at RO and R1.

        CALL POTENT (RO, R1, PO, DPO, P1, DP1)

        R(I)      = RO
        POTVAL(I) = PO
        IF (I .EQ. 999) THEN
          R(I+1)      = R1
          POTVAL(I+1) = P1
        ENDIF
10    CONTINUE

9999  RETURN
      END
```

```
C------------------------------------------------------------------------
C
C      Program:  SUBROUTINE PRNTER
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine prints the best set of potential
C                    parameters, the value of the potential function
C                    using these parameters, the observed and calculated
C                    energy levels with residuals.
C
C------------------------------------------------------------------------

      SUBROUTINE PRNTER (R, POTVAL)

      INTEGER I, J, N, NOEVAL, PARMNO, EVMLVL(25), EVMCNT,
     +        NELMT, IT1, IT2, IT3, IT4, IT5, IT6, STOP, JOBN

      REAL    EVAL(202), EVEC(202,202), H(20503),
     +        S(202,4), R(1000), POTVAL(1000), PARM(10), PARMLO(10),
     +        PARMHI(10), EVM(25), EVMW(25), RESID(25),
     +        STPSZE, BEGIN, CONST(10), HBAR, MU, L(202,4)

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                 NELMT, BEGIN, HBAR, MU
      COMMON /PARMS/   PARM, PARMLO, PARMHI, CONST, PARMNO
      COMMON /EIGENS/ EVAL, EVEC, NOEVAL, JOBN
      COMMON /MATRIX/ H, S, L, N

C     ------------------------------Rescale the parameters for display.

      DO 10 I=1,PARMNO
        PARM(I) = PARMLO(I) + PARM(I) * PARMHI(I)
10    CONTINUE

C     ------------------------------Print the best parameter set.

      WRITE (13,1301)
1301  FORMAT ('1The "best" set of potential energy parameters is:')

      WRITE (13,1302) (PARM(I),I=1,10)
1302  FORMAT ('0',5(G15.7),/,5(G15.7))

C     ------------------------------Print the value of the potential
C                                   function at 1000 grid points.

      WRITE (13,1303)
1303  FORMAT ('0','The values of the potential are:')
```

```
      DO 20 I=1,200
        WRITE (13,1304) R(I),POTVAL(I),R(I+200),POTVAL(I+200),
     +                  R(I+400),POTVAL(I+400),R(I+600),POTVAL(I+600),
     +                  R(I+800),POTVAL(I+800)
1304    FORMAT (' ',5('>',F7.4,G15.7,2X))
20    CONTINUE

C     ------------------------------Print the observed and calculated
C                                   energy levels and the residual.

      WRITE (13,1309)
1309  FORMAT ('1The weighted energies, eigenvalues, and residuals:')

      DO 25 I=1,25
        WRITE (13,1308) EVMLVL(I), EVMW(I), EVM(I), EVAL(I), RESID(I)
1308    FORMAT (' ',I2,2X,F7.5,3G15.7)
25    CONTINUE

C     ------------------------------Print the calculated energy levels

      WRITE (13,1305)
1305  FORMAT ('OThe calculated eigenvalues are:')

      STOP = 24
      IF (JOBN .NE. 0) STOP = N

      DO 40 I=0,24
        IT1  = 1 + I
        IT2  = 26 + I
        IT3  = 51 + I
        IT4  = 76 + I
        IT5  = 101 + I
        IT6  = 126 + I

        IF (STOP .LT. IT6) THEN
          IF (STOP .LT. IT5) THEN
        IF (STOP .LT. IT4) THEN
          IF (STOP .LT. IT3) THEN
            IF (STOP .LT. IT2) THEN
              IF (STOP .GE.IT1) THEN
                WRITE (13,1306) I, EVAL(I+1)
1306            FORMAT (' ',I3,2X,G15.7)
              ENDIF
            ELSE
              WRITE (13,1307) I, EVAL(I+1), (I+26), EVAL(I+27)
1307          FORMAT (' ',2(I3,2X,G15.7))
            ENDIF
          ELSE
            WRITE (13,1311) I, EVAL(I+1), (I+26), EVAL(I+27),
     +                      (I+51), EVAL(I+52)
1311        FORMAT (' ',3(I3,2X,G15.7))
          ENDIF
        ELSE
```

```
        WRITE (13,1310) I, EVAL(I+1), (I+26), EVAL(I+27),
     +                     (I+51), EVAL(I+52), (I+76), EVAL(I+77)
1310    FORMAT (' ',4(I3,2X,G15.7))
      ENDIF
      ELSE
        WRITE (13,1312) I, EVAL(I+1), (I+26), EVAL(I+27),
     +                       (I+51), EVAL(I+52), (I+76), EVAL(I+77),
     +                       (I+101), EVAL(I+102)
1312    FORMAT (' ',5(I3,2X,G15.7))
      ENDIF
    ELSE
      WRITE (13,1313) I, EVAL(I+1), (I+26), EVAL(I+26),
     +                       (I+51), EVAL(I+52), (I+76), EVAL(I+77),
     +                       (I+101), EVAL(I+102), (I+126), EVAL(I+127)
1313  FORMAT (' ',6(I3,2X,G15.7))
    ENDIF
40  CONTINUE

    IF (STOP .GE. 26) THEN
      I = 25
      WRITE (13,1306) I, EVAL(26)
    ENDIF

9999 RETURN
    END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE PLTRES
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  PLTRES creates a file (logical unit 15) containing
C                   the residuals of the least squares fit for plotting.
C
C-----------------------------------------------------------------------

      SUBROUTINE PLTRES

      INTEGER EVMCNT, EVMLVL(25), NELMT

      REAL    BEGIN, EVM(25), EVMW(25), RESID(25), STPSZE, HBAR, MU

      COMMON /ENERGY/  EVM, EVMW, RESID, EVMLVL, EVMCNT, STPSZE,
     +                 NELMT, BEGIN, HBAR, MU

C     ------------------------------Open the output file.

      OPEN (UNIT=15)

C     ------------------------------Write the vibrational quantum number
C                                   and the residual value to the file.

      DO 10 I=1,EVMCNT
        WRITE (15,1501) EVMLVL(I), RESID(I)
1501    FORMAT (' ',I3,G15.7)
10    CONTINUE

C     ------------------------------Write an End-Of-File mark and close
C                                   the output file.

      ENDFILE (UNIT=15)
      CLOSE (UNIT=15)

9999  RETURN
      END
```

```
C ---------------------------------------------------------------------
C
C     Program:  SUBROUTINE PLTPOT
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  PLTPOT creates a file containing the value of the
C                   potential energy model at 1000 grid points.
C
C---------------------------------------------------------------------

      SUBROUTINE PLTPOT (R, POTVAL)

      INTEGER I

      REAL    R(1000), POTVAL(1000)

C     ------------------------------Open the output file.

      OPEN (UNIT=16)

C     ------------------------------Write the grid position and the
C                                   value of the potential energy model.

      DO 10 I=1,1000
        WRITE (16,1601) R(I), POTVAL(I)
1601    FORMAT (' ',2G15.7)
10    CONTINUE

C     ------------------------------Write an End-Of-File mark and close
C                                   the file.

      ENDFILE (UNIT=16)
11    CLOSE (UNIT=16)

9999  RETURN
      END
```

```
C---------------------------------------------------------------------------
C
C      Program:  SUBROUTINE TRAILR
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  TRAILR closes the output listing file (logical
C                    units 13 and 6) and stops the CPU and wall time
C                    use statistics.
C
C---------------------------------------------------------------------------

      SUBROUTINE TRAILR

C     ------------------------------Shut down the run statistics.

      CALL ETIME
      CALL WTIME

C     ------------------------------Close the listing outputs.

      CLOSE (UNIT=13,STATUS='KEEP')
      CLOSE (UNIT=6)

9999  RETURN
      END
```

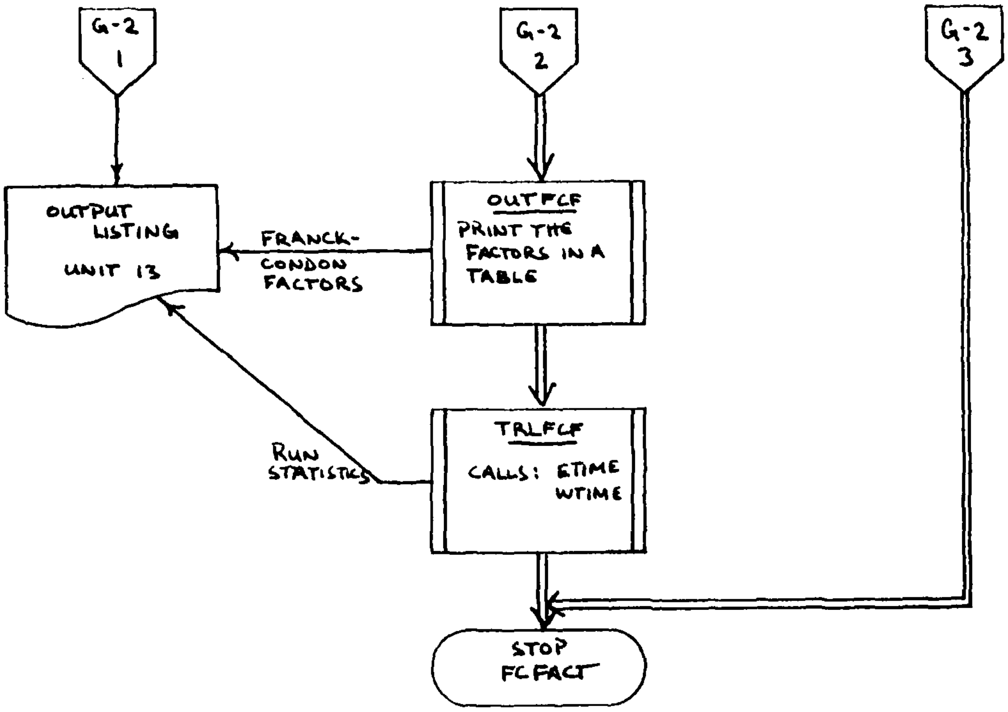
```
      ELSE
        WRITE (13,1302)
1302    FORMAT ('OERROR -- Wavefunction grids do not agree')
      ENDIF

C     ------------------------------Close the output file and run
C                                   statistics.

      CALL TRLFCF

9999  END
$ADD,HDRFCF
$ADD,RDRFCF
$ADD,CLCFCF
$ADD,OUTFCF
$ADD,TRLFCF
```

```
C      ------------------------------Read in all data concerning the
C                                    wave functions of the upper state
C                                    from logical unit 12.

      LUNIT=12
      CALL RDRFCF (LUNIT)

C      ------------------------------Make sure the grids used for both
C                                    states have the same number of grid
C                                    points.

      IF (NOPNTA .EQ. NOPNTB) THEN

C        ----------------------------Make sure both wave function sets
C                                    used identical S matricies.

        SOK = .TRUE.
        DO 50 I=1,NOPNTA
          IF (SA(I,1) .NE. SB(I,1)) SOK = .FALSE.
          IF (SA(I,2) .NE. SB(I,2)) SOK = .FALSE.
          IF (SA(I,3) .NE. SB(I,3)) SOK = .FALSE.
          IF (SA(I,4) .NE. SB(I,4)) SOK = .FALSE.
          IF (.NOT. SOK) GO TO 51
50      CONTINUE

C        ----------------------------Calculate the Franck-Condon factor
C                                    for every combination of the wave
C                                    functions from the two states and
C                                    store the value in FACTOR(LVLA,LVLB)

51      IF (SOK) THEN
          DO 200 I=0,24
            LVLA = I
            DO 100 J=0,24
              LVLB = J

C              ----------------------CLCFCF computes the Franck-Condon
C                                    factor for the two wave functions
C                                    PSIA(LVLA) and PSIB(LVLB).

              CALL CLCFCF (LVLA, LVLB)

100         CONTINUE
200       CONTINUE

C          --------------------------OUTFCF writes a table of Franck
C                                    -Condon factors to the listing.

          CALL OUTFCF

        ELSE
          WRITE (13,1301)
1301      FORMAT ('OERROR -- Matricies S do not match')
        ENDIF
```

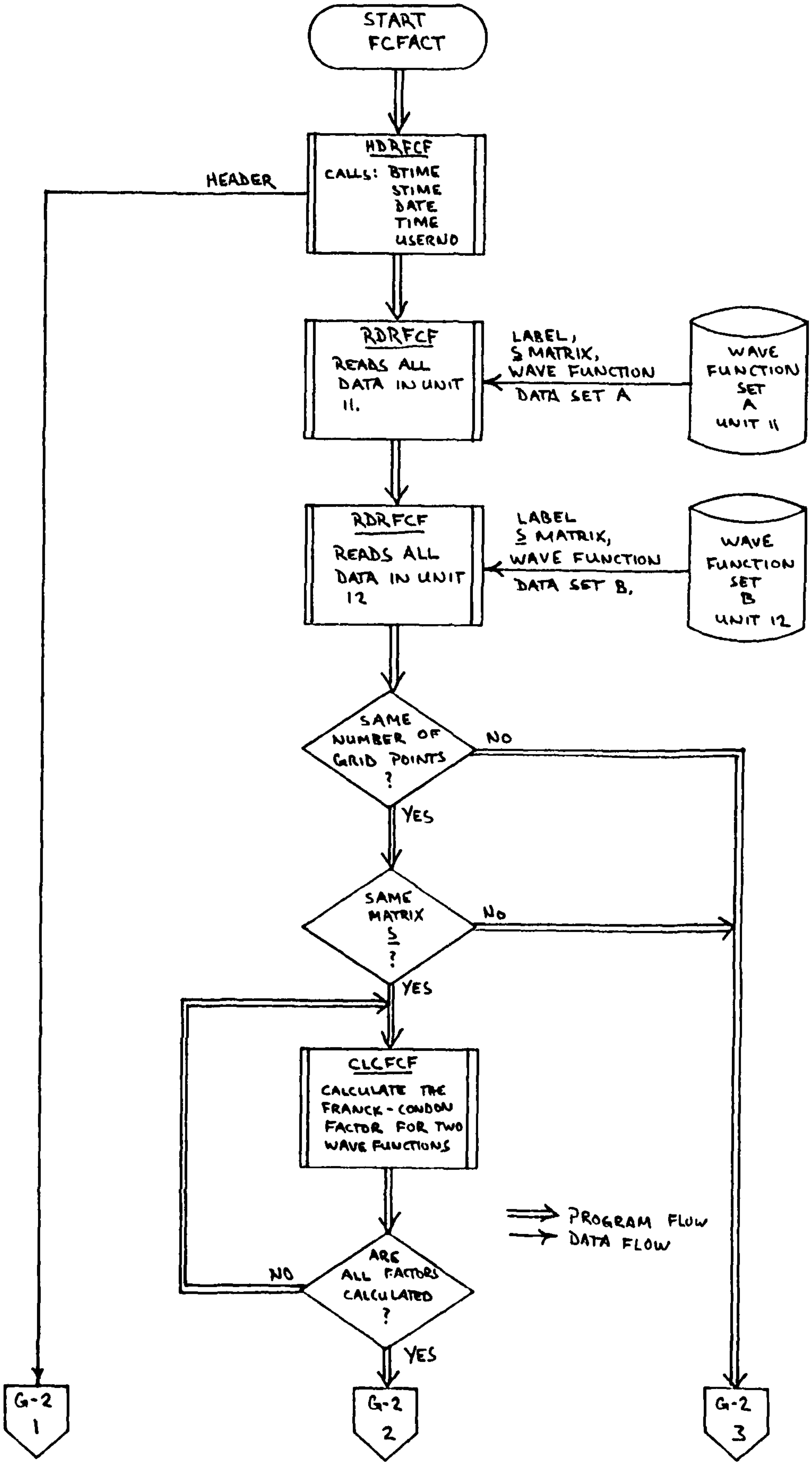
Appendix H

Program FCFACT

```
C-----------------------------------------------------------------------
C
C       Program:  FCFACT
C
C       Version:  84.11.30
C
C        Author:  Paul H. Ostdiek
C
C       Air Force Institute of Technology
C       Wright-Patterson Air Force Base, OH
C
C       Description:  This program calculates the square of the inner
C                     product between each wave function from two sets.
C                     This is the Franck-Condon factor.  Each wave
C                     function set is expected to have 25 (v=0 to 24)
C                     wave functions in the format used by DIATOM.
C
C                     I/O logical unit  6 -- output listing file
C                                      11 -- input file (wave set 1 v')
C                                      12 -- input file (wave set 2 v")
C                                      13 -- output listing file
C                                             (same as unit 6)
C
C-----------------------------------------------------------------------

      PROGRAM MAIN

      CHARACTER*72 LBLA, LBLB

      INTEGER I, J, LVLA, LVLB, LUNIT, NOPNTA, NOPNTB

      LOGICAL SOK

      REAL    FACTOR(0:24,0:24), PSIA(1:202,0:24), PSIB(1:202,0:24),
     +        SA(202,4), SB(202,4)

      COMMON /DTAFCF/ FACTOR, PSIA, PSIB, SA, SB, NOPNTA, NOPNTB
      COMMON /LABELS/ LBLA, LBLB

C     ------------------------------Open the output listing file, print
C                                   the header, and start run stats.

      CALL HDRFCF

C     ------------------------------Read in all data concerning the
C                                   wave functions of the lower state
C                                   from logical unit 11.

      LUNIT = 11
      CALL RDRFCF (LUNIT)
```


G-2
1
G-2
2
G-2
3
OUTPUT
LISTING
UNIT 13
FRANCK-
CONDON
FACTORS
OUTFCF
PRINT THE
FACTORS IN A
TABLE
TRLFCF
CALLS: ETIME
WTIME
RUN
STATISTICS
STOP
FCFACT

Appendix G
Program FCFACT Flow



```
        FIRST = .FALSE.
      ENDIF

      PREFIX = CONST(1) / (TP(2) - TP(1)) * -1.0
      RATIO  = CONST(2) / R1

C     ------------------------------Calculate the value of the
C                                   potential energy P1 and its slope
C                                   DP1 at the right edge R1 of the
C                                   grid element.

      P1  = TP(3)+(PREFIX*(TP(2)*(RATIO**TP(1))-TP(1)*(RATIO**TP(2))))
      DP1 = PREFIX*TP(1)*TP(2)/(TP(2)-TP(1))*(RATIO**TP(2)-RATIO**TP(1))

C     ------------------------------Limit infinity to a usable number.

      IF (P1 .GT. 1E10) THEN
        P1  = 1E10
        DP1 =-1E10
      ENDIF

C     ------------------------------Calculate the value of the
C                                   potential energy PO and its slope
C                                   DPO at the left grid edge RO.  Also
C                                   limit infinity.

      IF (RO .EQ. O) THEN
        IF (P1 .GE. 10000.0) THEN
          PO  = P1
          DPO = DP1
        ELSE
          PO  = 1E10
          DPO =-1E10
        ENDIF
      ELSE
        RATIO = CONST(2) / RO
      PO  = TP(3)+(PREFIX*(TP(2)*(RATIO**TP(1))-TP(1)*(RATIO**TP(2))))
      DPO = PREFIX*TP(1)*TP(2)/(TP(2)-TP(1))*(RATIO**TP(2)-RATIO**TP(1))

        IF (PO .GT. 1E10) THEN
          PO  = 1E10
          DPO =-1E10
        ENDIF
      ENDIF

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE POTENT (Mie)
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine returns the potential energy value and
C                   slope at the left and right grid element boundaries.
C                   The Mie function is used a model.
C
C-----------------------------------------------------------------------

      SUBROUTINE POTENT (R0, R1, P0, DP0, P1, DP1)

      INTEGER I, PARMNO

      LOGICAL FIRST

      REAL PARM(10), PARMLO(10), PARMHI(10), R0, R1, P0, DP0, P1, DP1,
     +     TP(10), CONST(10), PREFIX, RATIO, UPPER(10)

      COMMON /PARMS/  PARM, PARMLO, PARMHI, CONST, PARMNO

      DATA FIRST /.TRUE./

C     ------------------------------Scale the potential parameters PARM
C                                   to get correct parameter values TP.

      DO 10 I=1,PARMNO

        TP(I) = PARMLO(I) + PARM(I) * PARMHI(I)
        UPPER(I) = PARMLO(I) + PARMHI(I)

10    CONTINUE
      IF (FIRST) THEN
        WRITE (13,1301)
1301    FORMAT (/,/,'0Potential Model used is Mie')
        WRITE (13,1302)
1302    FORMAT ('02 Constants and 3 Parameters are used',/,
     +          ' Constant 1 is the Dissociation Energy',/,
     +          ' Constant 2 is the Internuclear Separation',/,
     +          ' Parameter 1 is the power alpha',/,
     +          ' Parameter 2 is the power beta',/,
     +          ' Parameter 3 is the energy shift')
        WRITE (13,1303)
1303    FORMAT ('0   Number         Constant          Parameter       L',
     +          'ower Limit      Upper Limit',/,' ------------  -',
     +          '--------------',3('  ---------------'))
        DO 20 I=1,10
          WRITE (13,1304) I, CONST(I), TP(I), PARMLO(I), UPPER(I)
1304      FORMAT (' ',5X,I2,5X,4(2X,G15.7))
20      CONTINUE
```

```
        FIRST = .FALSE.
      ENDIF

      PREFIX = CONST(1) * TP(2) / (TP(1) - TP(2))
      RATIO  = CONST(2) / R1
      IF (RATIO .GT. 6.5) RATIO = 6.5

C     -------------------------------Calculate the value of the
C                                    potential energy P1 and its slope
C                                    DP1 at the right edge R1 of the
C                                    grid element.

      P1   = TP(3)+(PREFIX*(RATIO**TP(1)-((TP(1)/TP(2))*RATIO**TP(2))))
      DP1 = PREFIX * (TP(1)/R1)*(RATIO**TP(2) - RATIO**TP(1))

C     -------------------------------Limit infinity to a usable number.

      IF (P1 .GT. 1E10) THEN
        P1   = 1E10
        DP1 =-1E10
      ENDIF

C     -------------------------------Calculate the value of the
C                                    potential energy PO and its slope
C                                    DPO at the left grid edge RO.  Also
C                                    limit infinity.

      IF (RO .EQ. 0) THEN
        IF (P1 .GE. 10000.0) THEN
          PO  = P1
          DPO = DP1
        ELSE
          PO  = 1E10
          DPO =-1E10
        ENDIF
      ELSE
        RATIO = CONST(2) / RO
        IF (RATIO .GT. 6.5) RATIO = 6.5
        PO  = TP(3)+(PREFIX*(RATIO**TP(1)-((TP(1)/TP(2))*RATIO**TP(2))))
        DPO = PREFIX*(TP(1)/RO)*(RATIO**TP(2)-RATIO**TP(1))

        IF (PO .GT. 1E10) THEN
          PO  = 1E10
          DPO =-1E10
        ENDIF
      ENDIF

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE POTENT (Lennard-Jones)
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine returns the potential energy value and
C                   slope at the left and right grid element boundaries.
C                   The Lennard-Jones model is used.
C
C-----------------------------------------------------------------------

      SUBROUTINE POTENT (R0, R1, P0, DP0, P1, DP1)

      INTEGER I, PARMNO

      LOGICAL FIRST

      REAL PARM(10), PARMLO(10), PARMHI(10), R0, R1, P0, DP0, P1, DP1,
     +     TP(10), CONST(10), PREFIX, RATIO, UPPER(10)

      COMMON /PARMS/  PARM, PARMLO, PARMHI, CONST, PARMNO

      DATA FIRST /.TRUE./

C     ------------------------------Scale the potential parameters PARM
C                                   to get correct parameter values TP.

      DO 10 I=1,PARMNO

        TP(I) = PARMLO(I) + PARM(I) * PARMHI(I)
        UPPER(I) = PARMLO(I) + PARMHI(I)

10    CONTINUE
      IF (FIRST) THEN
        WRITE (13,1301)
1301    FORMAT (/,/,'0Potential Model used is Lenard-Jones')
        WRITE (13,1302)
1302    FORMAT ('02 Constants and 3 Parameters are used',/,
     +          ' Constant 1 is the Dissociation Energy',/,
     +          ' Constant 2 is the Internuclear Separation',/,
     +          ' Parameter 1 is the power alpha',/,
     +          ' Parameter 2 is the power beta',/,
     +          ' Parameter 3 is the energy shift')
        WRITE (13,1303)
1303    FORMAT ('0   Number          Constant          Parameter         L',
     +          'ower Limit      Upper Limit',/,' ------------  -',
     +          '--------------',3('  ---------------'))
        DO 20 I=1,10
          WRITE (13,1304) I, CONST(I), TP(I), PARMLO(I), UPPER(I)
1304      FORMAT (' ',5X,I2,5X,4(2X,G15.7))
20      CONTINUE
```

```
        FIRST = .FALSE.
      ENDIF

      EXPNET = TP(1) * (CONST(2) - R1)
      PREFIX = 2 * CONST(1) * TP(1) * EXP(EXPNET)

C     ----------------------------/-Calculate the value of the
C                                   potential energy P1 and its slope
C                                   DP1 at the right edge R1 of the
C                                   grid element.

      P1  = CONST(1)*(((1-EXP(EXPNET))**2)-1)+TP(2)
      DP1 = PREFIX*(1-EXP(EXPNET))

C     ------------------------------Limit infinity to a usable number.

      IF (P1 .GT. 1E10) THEN
        P1  = 1E10
        DP1 =-1E10
      ENDIF

C     ------------------------------Calculate the value of the
C                                   potential energy PO and its slope
C                                   DPO at the left grid edge RO.  Also
C                                   limit infinity.

      IF (RO .EQ. 0) THEN
        IF (P1 .GE. 10000.0) THEN
          PO  = P1
          DPO = DP1
        ELSE
          PO  = 1E10
          DPO =-1E10
        ENDIF
      ELSE
        EXPNET = TP(1) * (CONST(2) - RO)
        PREFIX = 2 * CONST(1) * TP(1) * EXP(EXPNET)

        PO  = CONST(1)*(((1-EXP(EXPNET))**2)-1)+TP(2)
        DPO = PREFIX*(1-EXP(EXPNET))

        IF (PO .GT. 1E10) THEN
          PO  = 1E10
          DPO =-1E10
        ENDIF
      ENDIF

9999  RETURN
      END
```

```
C----------------------------------------------------------------------
C
C     Program:  SUBROUTINE POTENT (Morse)
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine returns the potential energy value and
C                   slope at the left and right grid element boundaries.
C                   The Morse function model is used.
C
C----------------------------------------------------------------------

      SUBROUTINE POTENT (RO, R1, PO, DPO, P1, DP1)

      INTEGER I, PARMNO

      LOGICAL FIRST

      REAL PARM(10), PARMLO(10), PARMHI(10), RO, R1, PO, DPO, P1, DP1,
     +     TP(10), CONST(10), PREFIX, EXPNET, UPPER(10)

      COMMON /PARMS/  PARM, PARMLO, PARMHI, CONST, PARMNO

      DATA FIRST /.TRUE./

C     ------------------------------Scale the potential parameters PARM
C                                   to get correct parameter values TP.

      DO 10 I=1,PARMNO

        TP(I) = PARMLO(I) + PARM(I) * PARMHI(I)
        UPPER(I) = PARMLO(I) + PARMHI(I)

10    CONTINUE
      IF (FIRST) THEN
        WRITE (13,1301)
1301    FORMAT (/,/,'OPotential Model used is Morse')
        WRITE (13,1302)
1302    FORMAT ('02 Constants and 2 Parameters are used',/,
     +          ' Constant 1 is the Dissociation Energy',/,
     +          ' Constant 2 is the Internuclear Separation',/,
     +          ' Parameter 1 is the factor beta',/,
     +          ' Parameter 2 is the energy shift')
        WRITE (13,1303)
1303    FORMAT ('0   Number         Constant           Parameter       L',
     +          'ower Limit      Upper Limit',/,' ------------  -',
     +          '--------------',3('  ---------------'))
        DO 20 I=1,10
          WRITE (13,1304) I, CONST(I), TP(I), PARMLO(I), UPPER(I)
1304      FORMAT (' ',5X,I2,5X,4(2X,G15.7))
20      CONTINUE
```

```
C       ------------------------------Calculate the value of the
C                                     potential energy P1 and its slope
C                                     DP1 at the right edge R1 of the
C                                     grid element.

        P1  = TP(1) * R1**2 + TP(2)
        DP1 = 2 * TP(1) * R1

C       ------------------------------Limit infinity to a usable number.

        IF (P1 .GT. 1E10) THEN
          P1  = 1E10
          DP1 =-1E10
        ENDIF

C       ------------------------------Calculate the value of the
C                                     potential energy PO and its slope
C                                     DPO at the left grid edge RO.  Also
C                                     limit infinity.

        PO  = TP(1) * RO**2 + TP(2)
        DPO = 2 * TP(1) * RO

9999    RETURN
        END
```

```
C-----------------------------------------------------------------
C
C     Program:  SUBROUTINE POTENT (Single Harmonic Oscillator)
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine returns the potential energy value and
C                   slope at the left and right grid element boundaries.
C                   The Harmonic Oscillator model is used.
C
C-----------------------------------------------------------------

      SUBROUTINE POTENT (R0, R1, P0, DP0, P1, DP1)

      INTEGER I, PARMNO

      LOGICAL FIRST

      REAL PARM(10), PARMLO(10), PARMHI(10), R0, R1, P0, DP0, P1, DP1,
     +     TP(10), CONST(10), PREFIX, EXPNET, UPPER(10)

      COMMON /PARMS/  PARM, PARMLO, PARMHI, CONST, PARMNO

      DATA FIRST /.TRUE./

C     ------------------------------Scale the potential parameters PARM
C                                   to get correct parameter values TP.

      DO 10 I=1,PARMNO

        TP(I) = PARMLO(I) + PARM(I) * PARMHI(I)
        UPPER(I) = PARMLO(I) + PARMHI(I)

10    CONTINUE
      IF (FIRST) THEN
        WRITE (13,1301)
1301    FORMAT (/,/,'0Potential Model used is Harmonic Oscillator')
        WRITE (13,1302)
1302    FORMAT ('00 Constants and 2 Parameters are used',/,
     +          ' Parameter 1 is the power alpha',/,
     +          ' Parameter 2 is the energy shift')
        WRITE (13,1303)
1303    FORMAT ('0   Number         Constant          Parameter       L',
     +          'ower Limit      Upper Limit',/,' ------------  -',
     +          '--------------',3('  ---------------'))
        DO 20 I=1,10
          WRITE (13,1304) I, CONST(I), TP(I), PARMLO(I), UPPER(I)
1304      FORMAT (' ',5X,I2,5X,4(2X,G15.7))
20      CONTINUE
        FIRST = .FALSE.
      ENDIF
```

Only one of the remaining four subroutines is used as subroutine POTENT. The following are routines for the Single Harmonic Oscillator, Morse, Lennard-Jones, and Mie potential energy models.

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE HDRFCF
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  HDRFCF opens the output listing file (logical units
C                   13 and 6) and starts CPU and wall time use
C                   statistics.
C
C-----------------------------------------------------------------------

      SUBROUTINE HDRFCF

      CHARACTER*8 VERSN
      CHARACTER*13 PCN

      INTEGER*3 IDATE(3), ITIME(3), IUSER(4)

      VERSN = '84.11.30'
      PCN   = 'GEP/84D-6/1.4'

C     ------------------------------Initiate the run statistics.

      CALL BTIME
      CALL STIME

C     ------------------------------Get the current date, time, and
C                                   user name for output on the header

      CALL DATE(IDATE)

      CALL TIME(ITIME)

      CALL USERNO(IUSER)

C     ------------------------------Open the output listing file and
C                                   write out the header

      OPEN (UNIT=13)
      OPEN (UNIT=6)

      WRITE (13,1301) IUSER, VERSN, IDATE, PCN, ITIME
 1301 FORMAT ('1 User:  ',4A3,T51,'Air Force Institute of Technology',
     *         T110,'Version:  ',A8,
     *       /,'  Date: ',3A3,T114,'PCN:  ',A13,
     *       /,'  Time: ',3A3,T57,'FRANCK-CONDON FACTORS',/,/)
```

```
C-----------------------------------------------------------------------
C
C      Program:  SUBROUTINE RDRFCF
C
C      Version:  84.11.30
C
C       Author:  Paul H. Ostdiek
C
C      Air Force Institute of Technology
C      Wright-Patterson Air Force Base, OH
C
C      Description:  This routine reads all records from one of two input
C                    files (LUNIT=11 or 12).  Each record read is written
C                    to the output listing (unit 13).  Data records are
C                    marked by a '>' in column 1.  These records contain
C                    data referenced by a single key word.  All other
C                    records are considered comments.  This routine uses
C                    an internal read, eg READ (SCORE(5:19),'(E15.7)') X
C                    reads from columns 5 to 19 of the character
C                    variable SCORE using the edit descriptor E15.7 into
C                    the real variable X.
C
C-----------------------------------------------------------------------

      SUBROUTINE RDRFCF (LUNIT)

      CHARACTER*72 LBLA, LBLB
      CHARACTER*80 SCORE

      INTEGER I, J, LUNIT, NOPNTA, NOPNTB, COUNT, TSTLVL, SCNT

      REAL    FACTOR(0:24,0:24), PSIA(1:202,0:24), PSIB(1:202,0:24),
     +        SA(202,4), SB(202,4)

      COMMON /DTAFCF/ FACTOR, PSIA, PSIB, SA, SB, NOPNTA, NOPNTB
      COMMON /LABELS/ LBLA, LBLB

C     ------------------------------Open logical unit LUNIT.

      OPEN (UNIT=LUNIT)

      DO 100 I=1,10000

C       ----------------------------Transfer a record from the input
C                                   file (unit 11 or 12) to the buffer
C                                   SCORE.

        READ (LUNIT,1101,END=101) SCORE
1101    FORMAT (A80)

C       ----------------------------If SCORE(1:1) is a '>' then SCORE
C                                   may contain data.
```

```
      IF (SCORE(1:1) .EQ. '>')THEN
        IF (SCORE(2:5) .EQ. 'LABL') THEN
C           ------------------------This record contains a label for the
C                                   state involved.

          IF (LUNIT .EQ. 11) LBLA(1:72) = SCORE(7:78)
          IF (LUNIT .EQ. 12) LBLB(1:72) = SCORE(7:78)
        ELSE
          IF (SCORE(2:2) .EQ. 'S') THEN

C             ----------------------This record contains data for the
C                                   S matrix.

            READ (SCORE(3:5),'(I3)') SCNT
            IF (LUNIT .EQ. 11) THEN
              READ (SCORE(6:20),'(E15.7)') SA(SCNT,1)
              READ (SCORE(21:35),'(E15.7)') SA(SCNT,2)
              READ (SCORE(36:50),'(E15.7)') SA(SCNT,3)
              READ (SCORE(51:65),'(E15.7)') SA(SCNT,4)
            ELSE
              READ (SCORE(6:20),'(E15.7)') SB(SCNT,1)
              READ (SCORE(21:35),'(E15.7)') SB(SCNT,2)
              READ (SCORE(36:50),'(E15.7)') SB(SCNT,3)
              READ (SCORE(51:65),'(E15.7)') SB(SCNT,4)
            ENDIF
          ELSE

C             ----------------------This record must have data for the
C                                   TSTLVL th wave function.

            READ (SCORE(2:5),'(I4)') TSTLVL
            READ (SCORE(6:9),'(I4)') COUNT
            IF (LUNIT .EQ. 11) THEN
              READ (SCORE(10:24),'(E15.7)') PSIA(COUNT,TSTLVL)
              IF (COUNT .GT. NOPNTA) NOPNTA = COUNT
            ELSE
              READ (SCORE(10:24),'(E15.7)') PSIB(COUNT,TSTLVL)
              IF (COUNT .GT. NOPNTB) NOPNTB = COUNT
            ENDIF
          ENDIF
        ENDIF
      ENDIF
100   CONTINUE

C     ------------------------------Close logical unit LUNIT.

101   CLOSE (UNIT=LUNIT)

9999  RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C     Program:  SUBROUTINE CLCFCF
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine calculates the Franck-Condon factor for
C                   the wave functions referenced by LVLA and LVLB.
C
C-----------------------------------------------------------------------

      SUBROUTINE CLCFCF (LVLA, LVLB)

      INTEGER I, J, LUNIT, NOPNTA, NOPNTB, LVLA, LVLB, IER,
     +        I1, I3, I202

      REAL    FACTOR(0:24,0:24), PSIA(1:202,0:24), PSIB(1:202,0:24),
     +        SA(202,4), SB(202,4), A(202), VA(202), VB(202),
     +        AREA

      COMMON /DTAFCF/ FACTOR, PSIA, PSIB, SA, SB, NOPNTA, NOPNTB

      DATA I1, I3, I202 /1, 3, 202/

      DO 100 I=1,NOPNTA
        VA(I) = PSIA(I,LVLA)
        VB(I) = PSIB(I,LVLB)
100   CONTINUE

      AREA = 0.0

C     ------------------------------Multiply SA*VA = A
C
C                                   VMULQF is an IMSL routine for matrix
C                                   multiplication.
C                                   (Band Symmetric storage mode times
C                                   Full storage mode.)

      CALL VMULQF (SA, NOPNTA, I3, I202, VA, I1, I202, A, I202)

C     ------------------------------Multiply VB*A = AREA
C
C                                   VMULFF is an IMSL routine for matrix
C                                   multiplication.
C                                   (Full storage mode times Full
C                                   storage mode.)
```

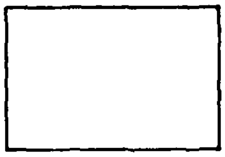
```
      CALL VMULFF (VB, A, I1, NOPNTA, I1, I1, I202, AREA, I1, IER)
C     -------------------------------The Franck-Condon factor is AREA
C                                    squared.
      FACTOR(LVLA,LVLB) = AREA**2
9999  RETURN
      END
```

```
C-----------------------------------------------------------------
C
C     Program:  SUBROUTINE OUTFCF
C
C     Version:  84.11.30
C
C      Author:  Paul H. Ostdiek
C
C     Air Force Institute of Technology
C     Wright-Patterson Air Force Base, OH
C
C     Description:  This routine prints a 25 by 25 Franck-Condon factor
C                   table (to unit 13).
C
C-----------------------------------------------------------------

      SUBROUTINE OUTFCF

      CHARACTER*72 LBLA, LBLB

      INTEGER I, J, NOPNTA, NOPNTB

      REAL    FACTOR(0:24,0:24), PSIA(1:202,0:24), PSIB(1:202,0:24),
     +        SA(202,4), SB(202,4)

      COMMON /DTAFCF/ FACTOR, PSIA, PSIB, SA, SB, NOPNTA, NOPNTB
      COMMON /LABELS/ LBLA, LBLB

      WRITE (13,1305) LBLA, LBLB
1305  FORMAT (' v" (across the page - lower state) is for ',/,
     +        ' ',A72,/,/,' v' (down the page - upper state) is for',/,
     +        ' ',A72)

      WRITE (13,1301) (J,J=0,15)
1301  FORMAT ('0','v'\v" ',I2,15(5X,I2),/)

C     ------------------------------Write the first part of FCF table.

      DO 10 I=0,24
        WRITE (13,1302) I, (FACTOR(J,I),J=0,15)
1302    FORMAT (' ',I2,1X,16(2X,F5.3))
10    CONTINUE

      WRITE (13,1303) (J,J=16,24)
1303  FORMAT ('1',1X,9(5X,I2),/)

C     ------------------------------Write the second part of FCF table.

      DO 20 I=0,24
        WRITE (13,1304) I, (FACTOR(J,I),J=16,24)
1304    FORMAT (' ',I2,1X,9(2X,F5.3))
20    CONTINUE

9999  RETURN
      END
```

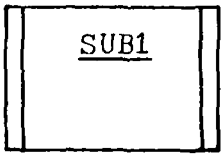
```
C----------------------------------------------------------------------
C
C       Program:  SUBROUTINE TRLFCF
C
C       Version:  84.11.30
C
C        Author:  Paul H. Ostdiek
C
C       Air Force Institute of Technology
C       Wright-Patterson Air Force Base, OH
C
C       Description:  TRLFCF closes the output listing file (logical
C                     units 13 and 6) and stops the CPU and wall time
C                     use statistics.
C
C----------------------------------------------------------------------

      SUBROUTINE TRLFCF

C     ------------------------------Shut down the run statistics.

      CALL ETIME
      CALL WTIME

C     ------------------------------Close the listing outputs.

      CLOSE (UNIT=13,STATUS='KEEP')
      CLOSE (UNIT=6)

9999  RETURN
      END
```

Appendix I

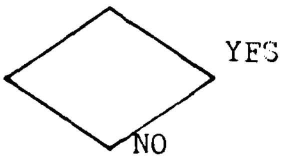
Program Flow Symbols



A process of some kind.

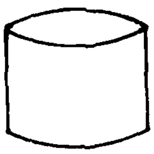


A module, e.g. SUB1 may be a subroutine or a function.

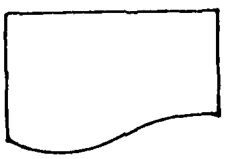


A decision point.

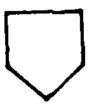
Start or stop a task.



A disk file.



Output listing (printer).



Off page connector

Appendix J

The Single Harmonic Oscillator

The Schrodinger wave equation describing a one dimensional, single harmonic oscillator is (10:75):

$$\frac{d^2\psi}{dx^2} + \frac{2\mu}{\hbar^2}(E - \tfrac{1}{2}kx^2)\psi = 0 \qquad \text{(J-1)}$$

where

$$\tfrac{1}{2}kx^2 = \tfrac{1}{2}\mu\omega^2x^2 \qquad \text{(J-2)}$$

Bounded solutions exist only for the discrete energy levels defined by the vibrational quantum number v and:

$$E_{(v)} = \hbar\omega(v + \tfrac{1}{2}) \qquad \text{(J-3)}$$

This equation shows that these energy levels are spaced equally apart.

The orthonormal wave functions of the harmonic oscillator are then described in terms of Hermite polynomials $H_v(\alpha^{\frac{1}{2}}x)$ as:

$$\Psi_{v(x)} = N_v e^{-\frac{1}{2}\alpha x^2} H_{v(\alpha^{\frac{1}{2}}x)} \qquad \text{(J-4)}$$

where the normalization constant N_v is

$$N_v = \left[\frac{1}{2^v v!}\left(\frac{\alpha}{\pi}\right)^{\frac{1}{2}}\right]^{\frac{1}{2}} \qquad \text{(J-5)}$$

and the scaling constant α is

$$\alpha = \frac{\mu\omega}{\hbar} \qquad \text{(J-6)}$$

The first ten Hermite polynomials are:

$$H_{0(q)} = 1$$

$$H_{1(q)} = 2q$$

$$H_{2(q)} = 4q^2 - 2$$

$$H_{3(q)} = 8q^3 - 12q$$

$$H_{4(q)} = 16q^4 - 48q^2 + 12 \qquad \text{(J-7)}$$

$$H_{5(q)} = 32q^5 - 160q^3 + 120q$$

$$H_{6(q)} = 64q^6 - 480q^4 + 720q^2 - 120$$

$$H_{7(q)} = 128q^7 - 1344q^5 + 3360q^3 - 1680q$$

$$H_{8(q)} = 256q^8 - 3584q^6 + 13440q^4 - 13440q^2 + 1680$$

$$H_{9(q)} = 512q^9 - 9216q^7 + 48384q^5 - 80640q^3 + 30240q$$

Vita

Paul H. Ostdiek was born in Ardmore, Oklahoma on 13 June 1957. He graduated from Park Hills High School, Fairborn, Ohio in 1975. He received appointments to the U.S. Air Force Academy, U.S. Naval Academy, and U.S. Coast Guard Academy that same year. In 1979 he graduated from the U.S. Air Force Academy, and was commissioned a second lieutenant in May of that year. His first assignment was to the principal laboratory of the Air Force Technical Applications Center at McClellan AFB. There he was a Scientific Data Systems Analyst his first year, and Chief, Resources, Computer Operations his second. During his third year he was Chief of the Microanalysis section. During his fourth year he designed and implemented a Scientific/Management Data Base System for the laboratory. In November 1981 he received the Air Force Commendation medal. He was selected Company Grade Officer of the Quarter for McClellan AFB in June, 1982. In June, 1983 he was awarded an oak leaf cluster to the Air Force Commendation medal, and entered the Air Force Institute of Technology.

Permanent address:
P.O. Box 123
Lawrence, Nebraska 68957

REPORT DOCUMENTATION PAGE

SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
Y CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
SIFICATION/DOWNGRADING SCHEDULE	
ING ORGANIZATION REPORT NUMBER(S) GEP/PH/84D-6	5. MONITORING ORGANIZATION REPORT NUMBER(S)
F PERFORMING ORGANIZATION ol of Engineering — 6b. OFFICE SYMBOL (If applicable) AFIT/EN	7a. NAME OF MONITORING ORGANIZATION
S (City, State and ZIP Code) 'orce Institute of Technology ıt-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
F FUNDING/SPONSORING IZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
S (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

Include Security Classification)
3ox 19

IAL AUTHOR(S)
H. Ostdiek, B.S., Capt, USAF

OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
1esis	FROM ____ TO ____	1984 December	189

MENTARY NOTATION

COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
GROUP	SUB. GR.	Franck-Condon Factors, Diatomic Molecules, Molecular Vibration, Schrodinger Equation, Numerical Analysis
04		
10		

ACT (Continue on reverse if necessary and identify by block number)

e: COMPUTER MODELING OF VIBRATIONAL ENERGY LEVELS OF POTENTIAL LASER CANDIDATES (DIATOMIC MOLECULES)

is Chairman: E. A. Dorko

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

BUTION/AVAILABILITY OF ABSTRACT IED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
. Dorko	513-255-4877	AFIT/ENP

SSIFIED

ASSIFICATION OF THIS PAGE

This thesis developed a finite element solution of the Schrodinger wave equation. This technique is used by a computer program to calculate the energy levels and wave functions of a diatomic molecule for a particular potential energy model. The potential energy model is a function of a set of parameters which a non-linear minimization routine varies before solving the wave equation. This is done in an iterative manner until the calculated energy levels agree in a least squares sense with the observed energy levels. Then the transition probabilities (Franck-Condon factors) between the wave functions are calculated by another program developed for this thesis. Finally, two programs were written to determine the energy levels observed in spectroscopic data. One uses Dunham coefficients and the Dunham equation while the second uses a least square fit to the data directly.

The four programs were tested and appear to work correctly. The numeric solutions were compared with the analytic solutions of the single harmonic oscillator. The lowest 25 energy levels agreed to within 0.005% accuracy while their wave functions appear to agree to within 0.40% accuracy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

END

FILMED

4-85

DTIC